SEVENTH FRAMEWORK PROGRAMME

Information & Communication Technologies
ICT

Cooperation Programme

Nippon-European Cyberdefense-Oriented Multilayer threat Analysis
[†]

Deliverable D1.4: Threat Data Final Report

Abstract: This deliverable compiles the final versions of the reports produced over the outputs of Tasks 1.1, 1.2 and 1.3, as well as the continuous tests run during Task 1.4. The final report will describe the complete data collection systems built by the Task 1.1. It will also provide the complete datasets built in Tasks 1.2 and 1.3, and make part of the datasets available to researchers outside of the consortia.

Contractual Date of Delivery	March 31, 2016
Actual Date of Delivery	April 20, 2016
Deliverable Dissemination Level	Public
Editor	Dawid Machnicki, Kenjiro Cho, and Yuji Sekiya
Contributors	All *NECOMA* partners

[†] The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7-ICT-2013-EU-Japan) under grant agreement n° 608533.

The *NECOMA* consortium consists of:

Institut Mines-Telecom	Coordinator	France
ATOS SPAIN SA	Principal Contractor	Spain
FORTH-ICS	Principal Contractor	Greece
NASK	Principal Contractor	Poland
6CURE SAS	Principal Contractor	France
Nara Institute of Science and Technology	Coordinator	Japan
IIJ - Innovation Institute	Principal Contractor	Japan
National Institute of Informatics	Principal Contractor	Japan
Keio University	Principal Contractor	Japan
The University of Tokyo	Principal Contractor	Japan

Contents

List of Figures

1 Introduction

This deliverable is a report of the achievements reached within workpackage 1 and tasks related to the activities from this work package. It mainly reprints materials from classified documents, presenting information that can be disclosed.

The main goal of workpackage 1 was to collect, transform and share the data. In particular, it aimed at providing a holistic view on the datasets which allowed conducting analyses on various logical layers in the project. But activities in workpackage 1 were not limited only to datasets investigation and gathering. They encompassed the knowledge management framework design, research on the best data exchange format to support the *NECOMA* architecture, and additional automated knowledge collection systems.

This document also provides specification of the datasets owned by the consortium members, for both, infrastructure and endpoint layers.

Lastly, it describes the results of research made possible by gaining access to the datasets.

1.1 Terminology

Throughout the document, the terms 'multilayer' and 'cross-layer' appear quite often as those terms directly relate to the core aspects of the project. In order to avoid ambiguity and misinterpretation, the definitions of those terms are provided below along with a diagram (Figure 1.1) illustrating how those terms fit in the *NECOMA* context in relation to the analysis modules:

- **multilayer:** pertaining to multiple layers, i.e. endpoint and infrastructure.
- **cross-layer:** bridging the border between layers, i.e. endpoint and infrastructure; implies multilayer.

Please note, that "multilayer" is a broader term than cross-layer. For example: In the context of threats, any attack on multiple layers is a multilayer attack (e.g. combining phishing with a simultaneous DDoS against the original site), but the term cross-layer attack requires using one layer to attack the other (e.g. using DNS spoofing to redirect to a phishing site or using a drive-by-download to build a botnet and perform a DDoS). In the context of threat analysis, processing multiple datasets from different layers always qualifies as multilayer analysis, even if separate tools and methods are used for each dataset, but a cross-layer analysis means processing multiple datasets from different layers in a single analysis with one or more anchors (such as IP addresses, time, ports) to bridge the link between the two.

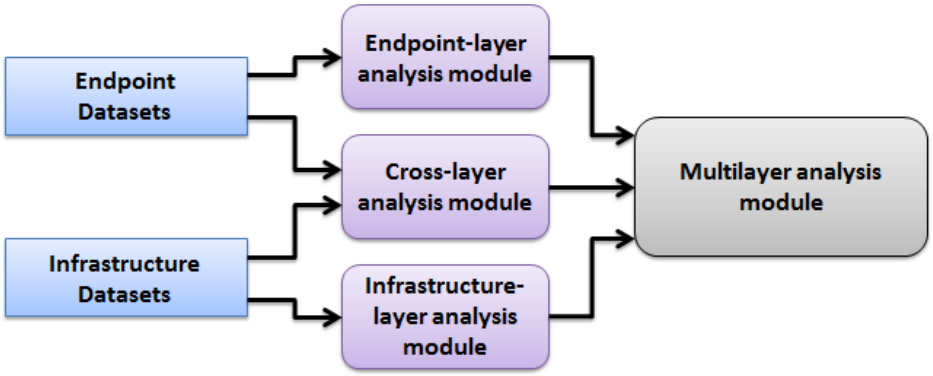


Figure 1.1: Cross-layer and Multilayer analysis modules relation

Other terms that appear are:

- **reconfiguration:** Replacing or modifying, partially or entirely, the configuration of a device or a system to make protected target(s) either more resistant to or more capable in mitigating the threat. e.g: change firewall filtering table, change SDN switch commutation table, deactivate LDAP account, etc.
- **resilience:** Ability of a system to maintain acceptable level of availability despite disruptions.

1.2 Structure of the Document

This deliverable follows the succeeding structure:

- **Chapter 1:** This section provides a description of the data collection infrastructure, starting from a survey of existing formats and data

sources, including all the data sources already under the control of partners of the project. The survey will assemble knowledge concerning existing mechanisms by investigating existing sources and formats. The survey outlines the coverage of existing collection mechanisms and try to specify new collection mechanisms and surfaces. Also a way to take advantage of the diverse datasets collected from various layers is defined. This section also provides a definition of a common data format, representing, organizing and structuring the raw data obtained from various heterogeneous sources. This format allows efficient and secure publication, exchange and storage of data for the purpose of dissemination and further analysis. Early data enrichment transforms existing data into an intermediary format closer to a common format. The new data format is specified with respects to existing and new data sources features.

- **Chapter 2:** This section shows the final, general architecture of the whole system. This architecture encompasses all the datasets, analysis, communication and resilience modules. Additionally it lists automated knowledge collection modules that were implemented in the systems final version.
- **Chapter 3:** Presents the outcomes based on datasets that, as a result, produced valuable academic papers, articles and other assets.

2 Collection Systems

The focus of this chapter encompasses components that are in the scope of workpackage 1. In relation to that, the structure of the chapter reflects this concept, dividing the document into three main chapters that are directly related to the datasets, APIs and the knowledge management system:

- **Common Data Exchange Format (2.1):** This section depicts the common data exchange format and API used in the project based on previous research. This section also presents a prototype implementation (2.1.3).
- **NECOMA General Architecture and Knowledge Management System Design (2.2):** This section presents the *NECOMA* system architecture on a more detailed level, based on previously specified requirements, recommendations and capabilities. It shows a detailed design and a proof-of-concept implementation of the Knowledge Management System component. Finally, this chapter outlines the design of automated knowledge gathering mechanisms that is part of the Knowledge Management System.
- **Automated Knowledge Collection (2.3):** This section presents the automated knowledge gathering components to assess the search abilities of different search engines as well as to which degree we are able to integrate them into our system. In addition to the usage of search engines our components are provided with the capabilities of automated ”web crawling” and extraction of relevant data. Those components serve to build rich data repositories that are integrated and shared within the system.

Each section begins with the specification of requirements or recommendations a particular component has to meet, derived from the vision of the

overall *NECOMA* system and its purpose. Then, it follows a study of state-of-the-art technologies and projects which may be reused, giving guidelines or ideas on how to approach and solve the problems outlined by the requirements. After the study is conducted, a working prototype is designed and developed which serves as proof of concept and a 'users manual', on how to approach the given challenges, for future development work.

2.1 Data Exchange Formats

Easy and effective data sharing among *NECOMA* project members required a common exchange method.

In this section, we introduce the common data exchange format designed and deployed in *NECOMA* project. The members of *NECOMA* project exchanged the datasets following the format and made analysis for threat detection.

Many well-defined data exchange formats and APIs have been proposed for this purpose. The selected data exchange format and API had to handle any data source and analysis module available in the *NECOMA* project. This chapter provides a comprehensive overview of existing solutions and introduces the data format and API that partners intend to employ.

2.1.1 Existing formats for exchange of security data

This section provides a study of the existing data objects and formats that have been studied with their usability to support multi-layer data exchange for the *NECOMA* system. The data formats were assessed regarding usability, flexibility and applicability in the *NECOMA* project context.

2.1.1.1 CybOX

CybOX (Cyber Observable eXpression) is intended to be a set of fundamental data types for communicating details of campaigns, threat sources, malware characteristics, and other cybersecurity events. As of this writing, with CybOX being an evolving specification, we consider CybOX version 2.1 in this document.

CybOX [1] has been specified as a set of rich and rigid XML schemas. In the context of the *NECOMA* project, using an XML schema may restrict the design space of research prototypes, as the project will deal with new kinds of threats that are not dealt with in current generation of products and services.

The *NECOMA* project will benefit from CybOX, since CybOX provides an extensive set of vocabularies for network-level and system-level observables.

Table 2.1: Example of data types in CybOX

Type of Observables	Data Type Names
Network level	AS, ARP_Cache, Address, DNS_Query, DNS_Record, Domain_Name, Email_Message, Hostname, Port, URI, Whois_Entry, X509_Certificate
System level	Account, Code_Object, Device, File, GUI_Dialogbox, Library, Mutex, Process, System, User_Session
Windows specific	Windows_Driver, Windows_Event, Windows_File, Windows_Handle, Windows_Registry_Key, Windows_Service, Windows_Thread
UNIX specific	Linux_Package, Unix_File, Unix_Pipe, Unix_User_Account, Unix_Volume

To give readers a concrete idea of the granularity and coverage of CybOX data types, some of the representative data types in CybOX have been presented in Table 2.1.

2.1.1.2 STIX and MAEC

STIX (Structured Threat Information eXpression) [4] and MAEC (Malware Attribute Enumeration and Characterization) [2] are two composite data structures that are built on top of CybOX. STIX conceptualizes and organizes different traits of threat information into independent and reusable constructs (data types). As of this writing, STIX being an evolving specification, we consider STIX version 1.0.1 in this document.

Currently, STIX is comprised of seven main constructs:

- **Campaign** is typically linked to a series of Incidents and can be linked to specific Threat Actors, since a Campaign typically employs specific TTP and individual Incidents can be linked with a common set of Indicators.
- **TTP** stands for tactics, techniques, and procedures; it can be used to link a series of Incidents to a specific Campaign.
- **Threat Actor** can be linked to a set of TTPs.
- **Incident** can be linked to a specific TTP used by a specific Threat Actor. During incident response, a set of Indicators may be identified. Incident can keep a record of the Course of Action that was taken during incident response.

- **Indicator** is typically linked to a set of relevant cyber observables. An Indicator can be linked to Course of Actions, or to TTPs.
- **Course of Action** characterizes both preventive measures and response measures.
- **Exploit Target** characterizes vulnerabilities or weaknesses that were targeted by specific TTPs. An Exploit Target may suggest a potential Course of Action, e.g., configuration changes or software upgrade.

MAEC specializes in communicating details of malware. MAEC conceptualizes malware at multiple levels of granularity – mechanism, behavior, action and its implementation – and bridges them through grouping primitives called bundle and package. Unlike traditional virus names that fail to convey deeper technical insight, MAEC tries to capture both behavior and mechanism of malware in a structured manner.

In the context of the *NECOMA* project, STIX provides a useful conceptualization of threat information. Threat analysis efforts in the *NECOMA* project benefit from its fundamental concepts and their associated vocabularies.

2.1.1.3 n6

The n6 platform [3] provides a simple REST API for data retrieval, which defines both query and response formats. The fundamental concept in the n6 data model is a security-related event that is described by a set of mandatory and optional attributes with well-defined semantics. Natively, events are represented as mostly flat JSON objects. However, additional output formats – CSV and IODEF [10] – are available to retain compatibility with other systems. Detailed specification of the API is presented in Section 2.1.2.

2.1.1.4 Data carrier and API for NECOMA

The two considered data carriers for the data exchange purpose were XML and JSON. For research purposes, JSON seemed to be the best option as a carrier of the data alleviating the need to prepare data binding libraries for each data type. Also, it significantly reduces the size of the carried message, increasing the performance while exchanging and parsing data.

The API chosen for *NECOMA* was n6. While more limited in scope than the formats discussed in the previous sections, one of the main design goals of the n6 API is to make integration as easy as possible from the client point of view. This property, combined with extendibility provided by JSON, made the API a good candidate for the exchange of heterogeneous datasets and it was chosen as the basis of the common data exchange mechanism in the *NECOMA* project. To accomplish that goal, the n6 API was extended

with additional attributes required to represent information present in all datasets used by the consortium.

Considering the rapidly evolving nature of the threat and the project mandate to deal with emerging threats, the *NECOMA* project had to minimize data-binding overhead in its implementation efforts. Nevertheless, conceptualizations and vocabularies that are designed and maintained by an international community of experts will be beneficial.

2.1.2 Specification of the n6 REST API

This section describes the n6 search interface. This is version 0.6.1 of the interface, which is the most recent at the time of writing. The n6 platform is under active development and new capabilities are added to the interface in a backwards-compatible way.

2.1.2.1 Overview

n6 uses an event-based data model for representation of all types of security information. Each event is natively represented as a JSON object with a set of mandatory and optional attributes (see "Event attributes" section below).

The REST API is available over TLS with mandatory authentication via client certificates. ABNF syntax of the generic URI scheme:

```
"https://" server "/" resource "." format "?" query
```

where

- `server` is a fully-qualified domain name of the API server
- `resource` is used to identify the desired scope of the data received, for the global dataset it must be set to `search/events`
- `format` is the requested format, can be `json`, `sjson`, `csv` or `iodef`
- `query` defines which events should be served (described in the next section)

2.1.2.2 Query

A query consists of a list of conditions on values of selected attributes. Query syntax in ABNF:

```
query = "?" arg *("&" arg)
arg = name "=" value
name = plain
value = plain / set
set =  plain *( "," plain )
```

where `plain` is percent-encoded (RFC 3986 [6]) character string. "Safe" characters that do not require encoding: `ALPHA / DIGIT / "-" / "." / "_" / "~"`, others must be encoded.

`name` corresponds to the name of the event attribute. Any attribute of string or numeric type can be used in queries. The name must be followed by the equals sign and the requested value of the attribute. Multiple values can be specified at the same time by separating them with commas or, alternatively, by repeating the attribute with different values.

Examples of complete URIs containing queries:

```
https://FQDN/RESOURCE.json?ip=10.0.0.1&modified.min=2016-01-01T00:00:00Z
https://FQDN/RESOURCE.json?name=%27%25xxx%27%3D
https://FQDN/RESOURCE.json?name=malware1,malware2
https://FQDN/RESOURCE.json?name=malware1&name=malware2
```

2.1.2.3 Response

n6 uses standard HTTP status codes: 200 (success), 206 (partial results), 400 (incorrect query), 403 (no permission), 405 (incorrect HTTP method), 416 (incorrect range request), 500 (server error).

Contents of the reply depend on the format requested, for JSON it is a single array where elements correspond to individual events. Each event is represented as a single JSON object with elements (keys) defined in the next section.

For large responses it is recommended to use "streamed" JSON variant (SJSON) which consists of concatenated top-level objects delimited by newlines (line feed, 0x10 ASCII). Each top-level object is represented in a single line (no pretty-print), which allows to parse results incrementally. Otherwise this format is identical with plain JSON.

In case of an error, only a text description is returned.

2.1.2.4 Event attributes

All attributes supported by the current version of n6 are listed below. [mandatory] denotes keys that must be present in their parent objects, by default all elements are optional. Element types are noted in brackets.

- action [string]
  Action taken by malware, e.g. redirect, screen grab.

- address [array of objects]
  Object containing IP address related to the threat. For malicious websites - A records in DNS, for connections to sinkhole and scanning hosts - source IP address. Elements of child objects:

  - ip [string] [mandatory]
    IPv4 address in dot-decimal notation.
  - ipv6 [string] [mandatory]
    IPv6 address in the hexadecimal notation. `ipv6` and `ip` are mutually exclusive - no more than a single address can be an element of the same object.
  - cc [string]
    Country code (ISO 3166-1 alpha-2).
  - asn [integer]
    Autonomous system number (without "AS" prefix).
  - dir [string]
    Role of the address in terms of the direction of the network flow in layers 3 or 4. Possible values: `src` (address is the source of the flow) / `dst` (address is the destination of the flow)
  - rdns [string]
    PTR record of the `.in-addr-arpa` domain associated with the IP address (without the terminal dot).

- adip [string]
  Anonymized destination address (see `dip`) in dot-decimal address without prefix, e.g. `x.184.216.119`.

- category [string] [mandatory]
  Category of the event. Possible values:

  - `amplifier`: hosts that can be used in amplification attacks (DoS)
  - `bots`: infected machines
  - `backdoor`: addresses of web shells or other types of backdoors installed on compromised servers
  - `cnc`: botnet controllers
  - `dns-query`: DNS queries and answers (no determination on legitimacy / maliciousness)
  - `dos-attacker`: (distributed) denial-of-service attacks - details related to sources
  - `dos-victim`: (distributed) denial-of-service attacks - details related to victims
  - `flow`: network traffic in layer 3 (no determination on legitimacy / maliciousness)
  - `flow-anomaly`: anomalous network activity (not necessarily malicious)
  - `fraud`: activities and entities related to financial fraud

  - `leak`: leaked credentials or personal data
  - `malurl`: malicious URLs (details about web servers infecting users)
  - `malware-action`: actions that malware is configured to make on infected machines
  - `phish`: phishing campaigns (similar to `malurl`)
  - `proxy`: open proxy servers
  - `sandbox-url`: URLs contacted by malware
  - `scanning`: hosts performing port scanning
  - `server-exploit`: attackers actively attempting to exploit servers
  - `spam`: hosts sending spam
  - `spam-url`: addresses found in spam
  - `tor`: Tor network nodes
  - `webinject`: injects used by banking trojans
  - `vulnerable`: addresses of vulnerable devices or services
  - `other`: other activities not included above

- confidence [string] [mandatory]
  Level of trust that the information is accurate. Possible values: `low` / `medium` / `high`

- count [integer]
  Connection (or other activity) count related to the event (applicable only to events resulting from aggregated data).

- dip [string]
  Destination IP address (e.g. sinkhole, honeypot) in dot-decimal notation. Does not apply to addresses of malicious websites.

- dport [integer]
  Destination port used in TCP or UDP communication.

- email [string]
  Email address associated with the threat (e.g. source of spam, victim of a data leak).

- expires [string]
  Time until the blacklist entry is considered valid.

- fqdn [string]
  Fully-qualified domain name related to the threat. For malicious websites - domain in the URL; for bots and scanners - destination domain.

- iban [string]
  International Bank Account Number associated with fraudulent activity.
- id [string] [mandatory]
  System-wide unique event identifier
- injects [array of objects]
  Objects describing a set of injects performed by banking trojans when a user loads a targeted website (see `url_pattern`). Exact structure of injects is dependent on malware family and not specified at this time.
- md5 [string]
  MD5 hash of the binary file related to the event.
- name [string]
  Category-dependent name of the threat, e.g. `virut`, `SSH Scan`.
- origin [string]
  Method used to obtain the data. Possible values:
  - `c2`: direct botnet controller observation
  - `dropzone`: botnet dropzone observation
  - `proxy`: monitoring traffic on a proxy server
  - `p2p-crawler`: active crawl of a peer-to-peer botnet
  - `p2p-drone`: passive listening to traffic in a peer-to-peer botnet
  - `sinkhole`: data obtained from sinkhole
  - `sandbox`: results from behavioural analysis
  - `honeypot`: interaction with honeypots, both client and server-side
  - `darknet`: monitoring of traffic collected by darknet
  - `av`: reports from anti-virus systems
  - `ids`: reports from intrusion detection and prevention systems
  - `waf`: reports from web application firewalls
- proto [string]
  Protocol used on top of the network layer: `tcp` / `udp` / `icmp`.
- restriction [string] [mandatory]
  Classification level, possible values: `internal` / `need-to-know` / `public`.
- sha1 [string]
  SHA1 hash of the binary file related to the event.

- source [string] [mandatory]
  Source (producer) of the event.

- sport [integer]
  Source port used in TCP or UDP communication.

- phone [string]
  Telephone number national or international. Consists of numbers, optionally prefixed by the plus symbol.

- registrar [string]
  Name of the domain registrar.

- status [string]
  Blacklist entry status. Possible values:

  - `active`: item currently in the list
  - `delisted`: item marked as inactive by an external source
  - `expired`: item is considered no longer active but might be still present in an external blacklist
  - `replaced`: some characteristics of an entry have changed and are represented as a new event (e.g. IP address change)

- replaces [string]
  Identifier (`id`) of the event that was superseded by the current one. Specific to blacklists.

- target [string]
  Organization or brand that is target of the attack (applicable to phishing).

- time [string] [mandatory]
  Time of the occurrence (not time of reporting), format defined in RFC 3339.

- until [string]
  Time of the last activity related to the event (applicable only to events resulting from aggregated data).

- url [string]
  URL related to the event, format defined in RFC 3986.

- url_pattern [string]
  Wildcard pattern or regular expression triggering injects used by banking trojans.

- username [string]
  Local identifier (login) of the affected user.
- x509fp_sha1 [string]
  SHA-1 fingerprint of an SSL certificate in hexadecimal format.

Attributes not listed above might appear in the results to represent source-specific data elements. The syntax and semantics of such attributes are not defined in this document.

Additionally, the following pseudo-attributes can be used in queries for specifying wider search criteria:

- url.sub [string]
  Substring in the `url` attribute.
- fqdn.sub [string]
  Substring in the `fqdn` attribute.
- ip.net [string]
  IPv4 network in CIDR notation, e.g. `10.0.0.0/8`.
- ipv6.net [string]
  IPv6 network, e.g. `2001:DB8::/32`.

A special class of pseudo-attributes are ones that refer to time ranges. Names of these attributes consists of two parts, where the first on defines data that is being queried:

- active [string]
  Refers to `time` and `expires` attributes: both are used for comparison and if either of them falls into the requested range, the whole criterion matches. E.g. `active.min=2014-10-04` would select events that either started after 2014-10-04 or started earlier but were still active after that date.
- modified [string]
  Time when data was made available through the API (e.g. time when the record was inserted into the internal database) or when content of an existing event has changed.
- time [string]
  Refers to the real `time` attribute.

The second part of the name of a pseudo-attribute consists of a one of the following operators:

- `.min` value is no earlier than the right-hand argument (inclusive)
- `.max` value is no later than the right-hand argument (inclusive)
- `.until` value is smaller than right-hand argument (exclusive)

2.1.2.5 Sample document in n6 format

Plain JSON format:


```
[
    {
        "address": [
            {
                "ip": "195.187.240.100",
                "cc": "PL",
                "asn": 12824
            }
        ],
        "adip": "x.2.137.140",
        "category": "bots",
        "confidence": "medium",
        "count": 18,
        "dport": 80,
        "fqdn": "example.com",
        "id": "26c8fd5097251dd15dc8431b267c65cf",
        "name": "B58-DGA2",
        "origin": "sinkhole",
        "proto": "tcp",
        "source": "b",
        "sport": 51869,
        "time": "2013-09-18T15:35:32",
        "until": "2013-09-18T19:00:00"
    },
    {
        "address": [
            {
                "cc": "PL",
                "ip": "108.162.201.25",
                "asn": 8308
            }
        ],
        "category": "malurl",
        "confidence": "low",
        "fqdn": "www.unknown-malware.eu",
        "id": "e1a53668ec9a2fe85974086815559868",
        "origin": "honeypot",
        "source": "m",
        "time": "2013-09-18T11:06:10",
        "url": "http://www.unknown-malware.eu/index.html?q=1"
```

```
    }
]
```

The same document in SJSON (lines truncated for readibility):

```
{"address":[{"ip":"195.187.240.100","cc":"PL","asn":12824}],"adip" ...
{"address":[{"cc":"PL","ip":"108.162.201.25","asn":8308}],"categor ...
```

2.1.3 Implementation of the n6 API for FORTH datasets

This section provides information about the common API for data exchange based on the already defined n6 format, as well as the description of the *NECOMA* dataset hosted at FORTH. Below follows a description of the sensors which contribute to the data collection process, some details of the implementation of a dataset server and the integration of the n6 API in it with a set of query examples. Although still under development, this prototype may already serve as a working proof-of-concept for the design principles taken so far and a guideline for future implementation work.

2.1.3.1 FORTH Sensors

The *NECOMA* dataset hosted at FORTH contains information gathered from a variety of sensors. These include the following: FORTH honey@home [1] honeypot data (AMUN deployment) which contains information on cyber-attacks gathered by monitoring end users' unused address space, data captured by Dionaea [2], another low-interaction honeypot deployment that captures attack payloads and malware, as well as data gathered from a set of publicly accessible web sources. Below are the sensors, from which data are currently collected, supported by the n6 server:

- BladeDefender [3]
- PhishTank [4]
- SANS [5]
- OffensiveComputing [6]
- Threat Expert [7]
- MD:PRO [8]

[1] honey@home project:http://www.honeyathome.org/
[2] Dionaea honeypot:http://dionaea.carnivore.it/
[3] BladeDefender: http://www.blade-defender.org/
[4] PhishTank: http://www.phishtank.com/
[5] SANS: http://www.sans.org/
[6] OffensiveComputing: http://www.offensivecomputing.net/
[7] Threat Expert: http://www.threatexpert.com
[8] MD:PRO: http://frame4.net/

2.1.3.2 Implementation of the n6 server

The n6 server is implemented entirely in Python using the `SQLAlchemy` toolkit to map the n6 queries with the corresponding database entities in the dataset, and `SimpleHTTPServer` module that handles the n6 queries (REST requests).

Currently, the preliminary implementation of the server does not support mandatory authentication via client certificates as described in the n6 documentation. Moreover, some event attributes should be revised in order to be consistent with the n6 format, e.g., the time attribute which has to follow the format defined in RFC 3339. At the moment, the time attribute is limited to a date string. Also, only the `json` output format is supported for the time being.

The fully-qualified domain name of the API server is: http://n6-necoma.ics.forth.gr/

2.1.3.3 Examples of using the FORTH dataset

Here, a number of examples using the FORTH dataset through n6 queries is presented:

Q1. Make a query for a suspicious IP address:

Query:
http://n6-necoma.ics.forth.gr/search/events.json?ip=74.125.79.99
Results:


```
[
   {
      "category":"spam",
      "origin":"honeypot",
      "confidence":"high",
      "latitude":"37.4192008972",
      "area code":"650",
      "region":"CA",
      "time":"2010-11-10",
      "asys":"GOOGLE - Google Inc.",
      "longitude":"-122.057403564",
      "source":"NoAH",
      "country name":"United States",
      "postal code":"94043",
      "dma code":"807",
      "country code":"US",
      "address":[
```

```
        {
           "cc":"US",
           "ip":"74.125.79.99",
           "asn":"15169"
        }
      ],
      "bgp prefix":"74.125.78.0/23",
      "allocated":"2007-03-13",
      "registry":"arin"
   }
]
```


The honeypots have found this IP address associated with spam activity.

Q2. Make a query for a suspicious MD5 hash:

Query:

http://n6-necoma.ics.forth.gr/search/events.json?md5=677daa8bf951ecce8eae7d7ee0301780

Results:


```
[
   {
      "category":"malicious-binary",
      "original-filename":"Net-Worm.Win32.Kido.js",
      "confidence":"high",
      "sha1":"879e553b472242f3ec5a7f9698bb44cad472ff3b",
      "name":"Net-Worm.Win32.Kido.js",
      "source":"Threat Expert",
      "time":"2009-10-31 23:59:46",
      "md5":"677daa8bf951ecce8eae7d7ee0301780",
      "origin":"web-sensor",
      "size":"119,296"
   },
   {
      "category":"malicious-binary",
      "original-filename":"677daa8bf951ecce8eae7d7ee0301780",
      "confidence":"high",
      "magic-file-type":"Win32.Worm.Downadup.A",
      "source":"Offensive Computing",
      "time":"2009-04-14 12:38:45",
      "origin":"web-sensor",
      "md5":"677daa8bf951ecce8eae7d7ee0301780"
   }
]
```

The requested MD5 is related with a malicious binary found on two web sources: Threat Expert and Offensive Computing.

Q3. Make a query for a suspicious URL:

Query:
http://n6-necoma.ics.forth.gr/search/events.json?url=http://www.swordmart.co.uk/images2/prjkt/

Results:


```
[
   {
      "category":"phish",
      "origin":"web-sensor",
      "confidence":"medium",
      "verified":"yes",
      "url":"http://www.swordmart.co.uk/images2/prjkt/",
      "online":"yes",
      "verification time":"2010-02-20T04:54:03+00:00",
      "source":"PhishTank",
      "phish detail url":"http://www.phishtank.com/phish detail.php?phish",
      "time":"2010-02-20T02:14:53+00:00"
   }
]
```


The requested URL has been found to be associated with phishing activity in the PhishTank web sensor.

Q4. Make a query that spans over 1 year:

Query:
http://n6-necoma.ics.forth.gr/search/events.json?time.min=2009-01-31&time.max=2010-01-31

Results:


```
[
   {
      "category":"malicious-binary",
      "origin":"web-sensor",
      "confidence":"high",
      "sha1":"4e7b4fa26d3b97a9532fa788059cff84639dcd19",
      "source":"Threat Expert",
      "time":"2010-01-13 05:16:51",
      "md5":"311d42cdf44e7e64a961d5a22639e64f",
      "size":"117,248"
```

```
   },
   {
      "category":"malicious-binary",
      "original-filename":"f2828f59fe4a56b07247289c6d4dd461",
      "confidence":"high",
      "sha1":"c8e9ed9d11d09e8f7eb3aa6174d7fcc3b2073bf3",
      "name":"Trojan-Downloader.Win32.FraudLoad.woxj",
      "magic-file-type":"(.EXE) Win32 Executable MS Visual C++",
      "md5":"f2828f59fe4a56b07247289c6d4dd461",
      "source":"MD:PRO",
      "time":"2009-11-29 17:06:04",
      "sha256":"6a337c0e8edafbad3ba96c7efa17db7d142f0ce629ec520f
       6def0dd9cd2740c2",
      "sha512":"4809250a87e4888caf4e072a63c9b28d2b619ce0c54cccf7
       b02539eebbaa023297ad1979f5f65ccf49331577c23ecf3615a6453e8
       204f3f3c2ed436f5a71a3"      origin":"web-sensor",
      "size":"1064484"
   },
   ...   {
      "registrant name":"Domain Administrator",
      "last updated at":"2009-07-07 00:00:00",
      "created at":"1995-01-18 00:00:00",
      "fqdn":"yahoo.com",
      "registrant email":"domainadmin@yahoo-inc.com",
      "source":"whois",
      "registrar":"Markmonitor.com",
      "expires at":"2012-01-18 00:00:00",
      "time":"2009-09-03"
   },
   {
      "registrant name":"Dns Admin",
      "last updated at":"2009-06-21 00:00:00",
      "created at":"1997-09-15 00:00:00",
      "fqdn":"google.com",
      "registrant email":"contact-admin@google.com",
      "source":"whois",
      "registrar":"Markmonitor.com",
      "expires at":"2011-09-13 00:00:00",
      "time":"2009-08-28"
   }
]
```

2.2 Knowledge Management Framework

Since the initial design of the system, the architecture was finalized to accommodate the requirements of the whole consortium. Additionally, the final design of the system evolved in time to address new challenges that were encountered in the course of the project.

The main focus was to facilitate data sharing and collective analysis, enabling more effective multilayer data correlation as well as better threat information sharing. Users and automated systems outside of the *NECOMA* platform can interact directly with analysis modules and receive results in a quick manner, at the same time enriching the *NECOMA*'s knowledge base.

2.2.1 Final design of the *NECOMA* system

Figure 2.1 shows the final architecture design. The structure of the system

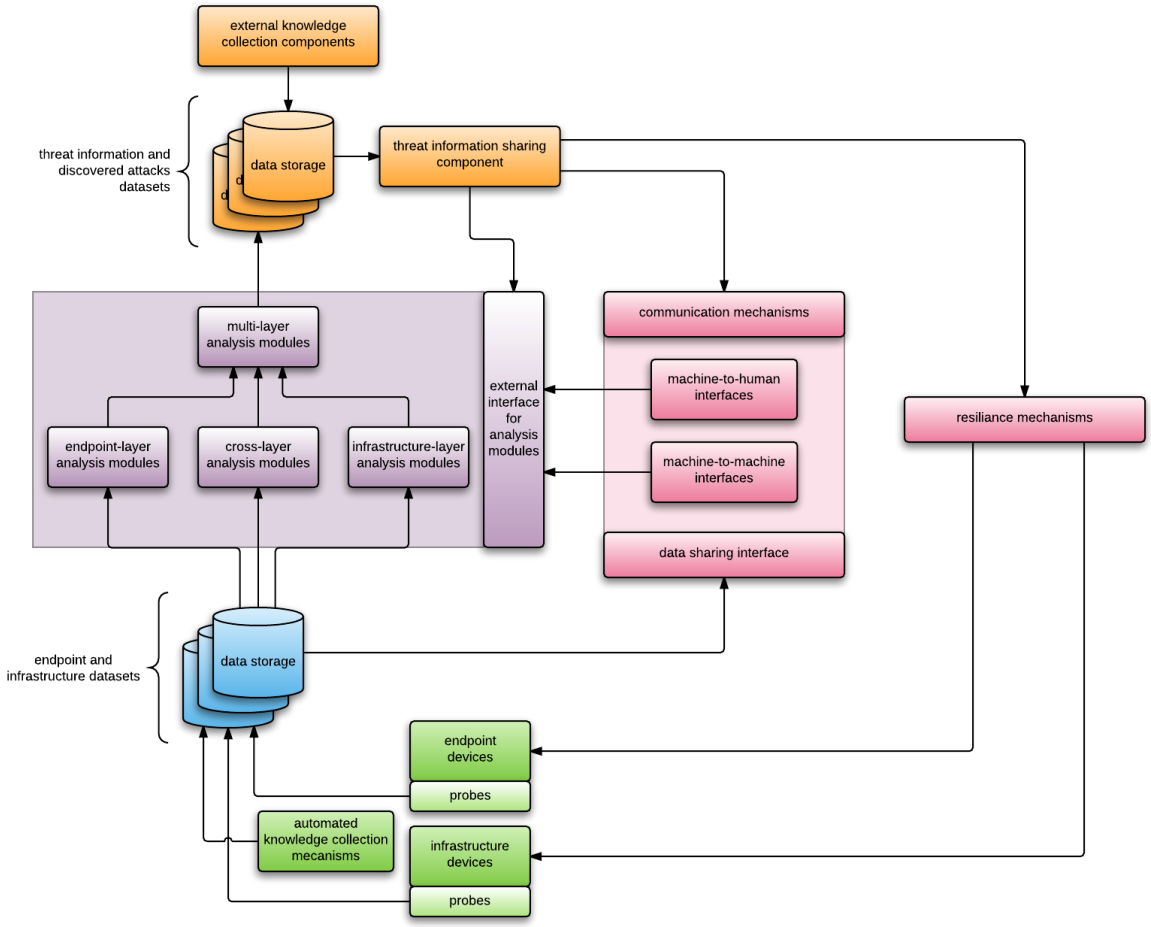


Figure 2.1: The final architecture of *NECOMA*

might be divided into three main areas:

- **Endpoint and infrastructure devices**
- **Analysis modules**
- **Communication mechanisms interfaces and resilience mechanisms**

Each of the items will be explained in the following chapters.

2.2.2 Endpoint and infrastructure devices

The Endpoint and Infrastructure devices are both at the beginning and the end of the processing pipeline in our system. They comprise both sources of raw data, to be processed later, and the assets to be protected or re-configured when an attack is discovered. In *NECOMA*, we distinguish two main data layers which are then divided into five source categories each. Table 2.2 lists all the data sources[9] and types which contribute to building the *NECOMA*'s data storage.

Endpoint Layer
Mail and Messaging Dataset
Web Dataset
User Behaviour Dataset
Sinkhole Dataset
Client Honeypots and Sandbox Dataset
Infrastructure Layer
Traffic dataset
DNS Dataset
Topology Dataset
Telescope Dataset
Early Warning Dataset

Table 2.2: Data sources.

Members of the consortium have contributed a total of 34 datasets, out of which 29 are still being constantly expanded through active collection. The consortium has worked on means to efficiently share the data sets with external actors. As the result of those actions, 20 datasets have been made available. Table 2.3 summarizes the statistics of datasets in control of the consortium.

[9] For details about the data sources and data sets, please refer to Deliverables D1.2: *Infrastructure-layer threat data sets* and D1.3: *Endpoint-layer threat data sets*.

	Infrastructure Layer	Endpoint Layer
Total number of datasets	25	9
Ongoing capture	25	5
Estimated size	23,7 TB	200 GB
Shared (non-private)	15	5

Table 2.3: Data sets statistics.

Additionally, the system's data collection capabilities do encompass Automated Knowledge Collection Mechanisms. The experiments and initial, working prototypes proved that the datasets can be easily enriched by automated mechanisms such as web crawlers, collecting the contents of suspicious web pages, and also the utilization of search engines for seeking information related to computer security published openly on the Internet.

Research activities in *NECOMA* did also encompass creating a unified, common data storage, that would be a single interface for storing and accessing all the captured, raw information. Although such a design was proposed, the complexity and requirements (including law and regulations apart from technical requirements) made the idea to reach far beyond *NECOMA*'s scope.

2.2.3 Analysis modules and threat information sharing

To take advantage of the huge amounts of captured data, *NECOMA* devoted significant efforts to design and implement advanced analysis modules that are able to produce actionable information on contemporary threats. This information is distributed through External Interfaces and utilized by Resilience Mechanisms, which will be described in the following section.

The analysis modules can be divided into three main categories:

- **Infrastructure-layer analysis modules**
- **Endpoint-layer analysis modules**
- **Cross-layer analysis modules**

The architecture diagram depicts yet another type of module: multilayer analysis module. This term was created as an abstract node for structuring and correlating outcomes coming from various analysis modules that are not intended to interact. In most cases the analysis results are simply forwarded, but in the context of threats affecting multiple layers (e.g., combining phishing with a simultaneous DDoS against the original site), additional correlation may be performed and results redirected for further processing.

A total of sixteen analysis modules have been implemented within the scope of *NECOMA*. Detailed descriptions of the modules are provided in the

Chapter 3 of D2.2: *Threat Analysis Platform*, while D2.1: *Threat Analysis* contains an in-depth description of the underlying techniques.

The analysis modules work on the datasets collected by the consortium, although several modules expose input interfaces that enable direct interaction with external actors. For example, external users can submit suspicious URLs to the phishing detection modules in order to assess the credibility of websites. Such scenarios will be covered in the following section.

Another significant component is the *threat data storage*, accessed through the threat information sharing component. Conceptually, the threat data storage component is a single dataset containing output generated by all analysis modules. It holds information about malicious activities, malware and, in general terms, any kind of valuable information that can be extracted from the analysis results and reused by resilience mechanisms or external actors. It consists of multiple databases under the control of *NECOMA*'s consortium members that expose a common interface. Additionally, in order to further enrich produced information, *NECOMA* takes advantage of external threat data sources, such as Phishtank.

To enhance multilayer analysis, the analysis modules are capable of accessing the threat data storage and taking advantage of the collected threat knowledge. This allows for a much broader view of the threat landscape and more effective correlation of analysis results coming from different modules and also external interfaces and sources. This loop further enriches the threat knowledge and may lead to discovering much more sophisticated multilayer attacks.

In order to facilitate the communication between various interfaces within the system and to enable easy access for external actors, the analysis modules as well as the data storages implement the n6 API[10]. By using the n6 API, *NECOMA* implements a unified way for inter-component communication integrating the numerous components tightly at the same time allowing flexibility in extending the system with new data sets and analysis modules. Furthermore, it provides an easy and documented way for interacting with the analysis modules and datasets.

2.2.4 Communication mechanisms and resilience mechanisms

The last stage in the *NECOMA* processing pipeline are the modules that utilize the threat knowledge produced during analysis. They can be divided into two major groups: *resilience mechanisms* and *communication mechanisms*.

Resilience mechanisms consist of all elements of the protected network, system, or application that can trigger reconfiguration of a device

[10] The implementation tutorial can be found in Deliverable D3.2: *Security Information Exchange – Design*, section 3.

in response to an attack. Two main categories of resilience mechanisms can be distinguished: mechanisms for the endpoint layer and for the infrastructure layer[11]. Both types of defences directly utilize the available threat knowledge and are capable of reconfiguring devices settings in order to mitigate an attack (reactive) or prevent a threat (preventive). *NECOMA* focuses mostly on the reactive measures, although the need for secure by design mechanisms is strongly highlighted and expected as a follow-up result of the project.

Communication mechanisms serve the purpose of information exchange between external actors and the system and, enrichment of the system's knowledge. They facilitate dissemination of information collected in the *NECOMA* platform with the goal of utilizing it outside of the system. Communication mechanisms are also used to provide access to functionality offered by analysis modules to external users.

2.2.5 Knowledge management system architecture

The knowledge management system for the *NECOMA* project provides an *information pipeline* from the **threat data collection**. It tries to leverage past and current work on the topic with the goal of expanding these existing mechanisms and orienting them towards threat data analysis. Then, **threat data analysis** is considered, not only from the perspective of understanding attackers and vulnerabilities, but also from the point of view of the target and victim, having the need to protect himself in real-time and in the most efficient manner possible.

2.2.5.1 MATATABI: implementation of analysis platform

While the previous section introduced the overall design of the *NECOMA* platform, this section describes the implemented prototype – MATATABI – which connects all elements of the architecture to provide a complete security information processing pipeline.

MATATABI is built upon the Apache Hadoop framework in order to fulfill some requirements: 1) scalability, 2) real-time analysis, and 3) uniform programmability [23]. The implementation covers the functionalities established by the *NECOMA*'s architecture, including interfaces to external entities such as human analysts or automated systems using results of processing modules. Figure 2.2 provides a high-level overview of the system, each component will be described in the following sections.

[11] An extensive design of those mechanisms is available in Deliverable D3.4: *Countermeasure Application – Design*.

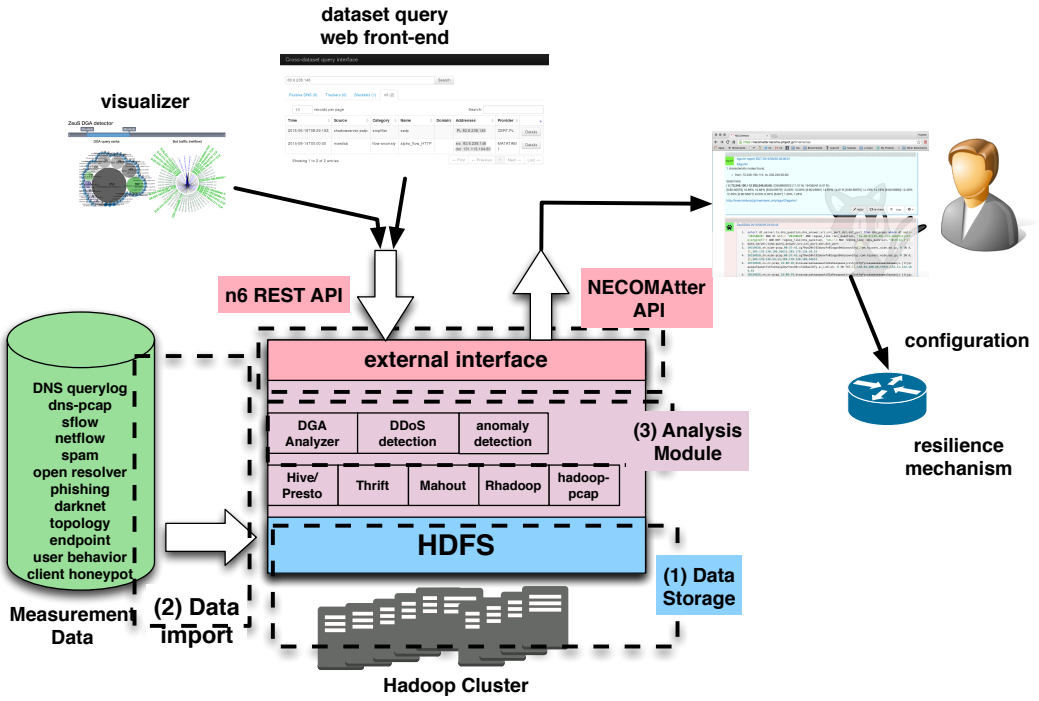


Figure 2.2: Core elements of the *NECOMA* architecture implemented by MATATABI: *data probe*, storage, analysis modules, external interfaces. Coloring of elements corresponds to figure 2.1.

2.2.6 Endpoint and infrastructure devices

The data import component of MATATABI collects the data at various devices (routers, DNS servers, and crawler) as *data probes*. Those probes are located at selected measurement points in the infrastructure and store all collected data in a distributed filesystem, which is a part of a Hadoop instance (*data storage*).

2.2.7 Analysis modules and threat information sharing

Once the data is collected, analysis modules try to look for security threats. MATATABI uses a simple programming model with a powerful computation backend to sift through the huge amount of data of different kinds, which allows to easily implement cross- and multi- layer analysis (*analysis module*).

2.2.8 External interfaces and resilience mechanisms

Results of analyses are accessible through an application programming interface (API) implemented using the n6 SDK (common *machine-to-machine*

interface) and through the NECOMAtter API together with its associated web front-end (*machine-to-human interface*).

NECOMAtter also has the ability to control external entities such as resilience mechanism in the *NECOMA* architecture. DDoS mitigation is one of the use cases: results of analyses are reported as *tweets* through NECOMAtter and an application acting as a resilience mechanism executes a command based on the tweeted information, which eventually reconfigures access control lists of Open vSwitches. Since fully automated operation might be too risky in a production environment, reconfiguration can be done by human operators with the help of a machine-to-human interface provided by NECOMAtter.

2.3 Automated Knowledge Collection

The previous section provided the final design of our comprehensive and broad knowledge management system that encompasses the datasets available in the project. Although very extensive, the datasets within the system may not be sufficient to allow accurate assessment of potential threats. In order to extend our system's data aggregation capabilities, we take advantage of existing search engines that are able to efficiently browse external web sources.

This section describes research activities that were carried out in order to assess the search abilities of different search engines as well as to which degree we would be able to integrate them into our system through automated knowledge gathering components. In addition to the usage of search engines, our components will be provided with the capabilities of automated "web crawling" and extraction of relevant data. Those components will serve to build rich data repositories that will be integrated and shared within the system.

2.3.1 R-LING web crawler

The main purpose of the R-LING web crawler is automated browsing through the web, oriented towards identifying and reporting phishing web sites. It can extract data from publicly available web sites that may be relevant for assessing any phishing intention. It is also capable of automated ranking of the visited web sites, in terms of their phishing intent, automatically building a rich dataset that provides web site phishing ranking and possible links with other sites and endpoints.

The R-LING component consists of two main modules:

- Standalone server

- Database

The standalone server is the core part of the component and is responsible for data gathering and analysis activities. It also allows other components and users to interact with the R-LING dataset. The standalone server module is divided into four modules:

- Crawler module
- Scouting module
- Ranking module
- Communication and UI module

The **Crawler module** is responsible for data gathering and extraction. First, it queries the R-LING database to obtain URLs and web resources that were not yet scanned in terms of suspicious contents. Once the URL is obtained, the *Crawler module* sends a request to the URL and analyses the response extracting information like URLs, specific text phrases occurrences and associated HTML tags. When scanning of the response has completed, the data is stored in the database along with additional meta-data and eventual links to other instances of data related to the obtained information.

The **Scouting module** aims at checking if the URL obtained was already analyzed (or identified as malicious) by any of the internal *NECOMA* components. This module will make use of the n6 API as well as the knowledge management system.

If any information, regarding the obtained URL, is not found internally within the *NECOMA* system, the *Scouting module* will try to use external sources to find any possible occurrence of the URL and gather as much related information as possible.

If no information regarding the URL was found (internally or externally to the *NECOMA* system), the **Ranking module** will take advantage of several analysis algorithms in order to rank the data found in the R-LING database. The term “rank” is used here, since we try to avoid false positives as much a possible. Thus, the algorithms are used to analyse the information previously extracted form web resources and give a “likelihood indication” on whether the web resources are phishing contents or not. The algorithms analyse such things as domain names, SSL certificates, extracted parts of the response contents also trying to link all that information with legitimate sites in order to reveal impersonation attempts.

The **Communication and UI module** aims at sharing the R-LING database with other *NECOMA* components and potential (human) users through a UI.

The component will implement the n6 API as a web service that will run on top of the database enabling external communication. The UI will allow adding additional information to the database manually and enable inspection of the results that may be found in the R-LING database.

The R-LING Database is a non-relational database that stores JSON documents. It holds the results of the analyzed data along with ranking information, analysis history and relations between analyzed information. It may be easily extended with additional information if needed, also by storing correlation information and analysis results coming from external components for further analysis.

The R-LING architecture is not limited to the existing segments and may be extended thanks to the flexible nature of technologies in which the R-LING component is implemented. Furthermore, the segments are standalone, separate applications which may exist and function autonomously.

We are currently assessing the possibilities of using external search engines for the R-LING Scouting module. The primary focus is on the leading and popular brands, since those can provide the most comprehensive results while querying for information, although the less popular ones are not excluded. Initial investigation shows difficulties since many search engines require a licence purchase in order to take full advantage of their browsing capabilities and the API they expose. On top of that, the licences are usually based on the volume of generated queries, and given the amount of data that would have to be processed, this possibility is excluded. However, free licences might provide sufficient functionalities to satisfy demonstration requirements.

Another possibility is to query directly the globally exposed API, such as *google.com*, but this will require further study as it may violate the terms of use of particular vendors.

Additionally, we are constantly improving the ranking algorithms used by the *Ranking Segment*, investigating different approaches on phishing site analyses. The result of these research activities will be described in detail in the workpackage 2.

2.3.2 Tokenseeker tool for searching web resources

Since a lot of information related to computer security is published openly on the Internet, it is relatively easy to find by leveraging the fact that search engines index the majority of publicly available content. By querying these engines, web pages containing tokens (keywords) of interest may be found. We created a simple tool to automate this process – *tokenseeker*.

The architecture of the tool is straightforward: for each data object (e.g. in the n6 format, described in section 2.1.2) given as input, it extracts one or more attributes and performs queries against predefined search engines, in attempt to find related information on the web. Results obtained from search engines are summarized and returned on output in JSON format.

Two different methods to retrieve information from the web are applied: *direct* and *contextual* queries. By *direct query* we mean queries where only attributes extracted from the input data are used for searches. Examples of such attributes include IP addresses, MD5 hashes, domains, etc. In a *direct query*, a single attribute can be used or, to obtain more specific results, a combination of attributes are used in conjunction. *Contextual queries* work in the same way, except that an additional phrase is added to the set of searched keywords, so the results are further narrowed down to web pages containing the requested phrase. For example, by adding 'malware' or 'infect' phrase to a query, we can select web pages that most likely describe malicious behavior. Introduction of contexts can help finding important data when a direct query returns too many results for a given token, e.g. for the domain of a popular web site.

Results returned by the tool allow to obtain the following information:

- if the set of tokens was indexed by a search engine at all,
- approximate number of web sites containing the tokens,
- what contexts was the token used in,
- if any established sites (e.g. major news) referred to the tokens,

URLs leading to found web pages are returned as well, although they are not followed automatically by tokenseeker – this is an auxiliary information that may be presented to users of the tool during interactive operation. Future work include adding a crawler component that would visit web pages and extract interesting data in a form suitable for further machine processing.

Tokenseeker can also be used to identify vulnerable web sites or malicious contents (e.g. exploit kits) that have been indexed by search engines. In order to accomplish that, input for the tool consists of text patterns used to identify such sites and the parameter limiting the number of URLs fetched from search engines is usually set to a large number to get a complete list, if possible. Ready-to-use patterns are published publicly, most notably at Google Hacking Database [12], however from our experiments they often return many false positives, therefore manual testing and tuning is required in each case.

[12] Google Hacking Database: http://www.exploit-db.com/google-dorks/

After the preliminary evaluation of search engines for use with tokenseeker, which resulted in choosing two well established services: Microsoft's Bing and Google Search. Both provide comprehensive REST APIs, with a certain quota of free queries and paid subscription options. Initial tests confirmed that their databases contain up-to-date indexes of sites with computer security news. One advantage of Google is better availability of ready to use queries – for example all queries in GHDB are tested only with Google and some of them do not provide expected results after being ported to the Bing query syntax. However, a major problem with Google that we encountered is unavailability of a global search API which was present in older versions[13] – now all queries must use Custom Search Engine or Site Search services. Out of these two, only CSE can be configured to search the entire internet,[14] however the API provides only the first 100 results for a given query, which limits the ability to perform exhaustive searches.

Other engines were rejected for several reasons, most importantly many niche engines do not have enough indexed content, while some major ones lack search API (DuckDuckGo, Baidu). Some engines that required up-front payment (Yahoo) or complex registration process for developers (Yandex) were not tested due to time and resource constraints – experimenting with them is left as future work.

2.3.3 Crawler for phishing websites

Phishing is a cyber threat that attempts to acquire personal information by tricking an individual into believing that the attacker is a trustworthy entity. Phishing attackers lure people by using a phishing email, as if it were sent by a legitimate corporation. The attackers also attract the email recipients into a phishing site, which is the replica of an existing web page, to fool a user into submitting personal, financial, and/or password data.

We explored automated data collection systems to capture new phishing sites. The biggest open resource is PhishTank [20]. PhishTank.com is operated by OpenDNS [21] who provides a DNS resolution service for consumers and businesses as an alternative to using their Internet service provider's DNS servers. Actually, PhishTank accepts free submission of suspect URLs considered to be phishing sites. The submission is limited to registered users of PhishTank.com, but anyone can register to it. Due to the possibility to submit phishing candidates arbitrarily to PhishTank.com, some sites are mistakenly reported as phishing sites. In order to improve the reliability of anti-phishing databases, the reported URLs are validated by registered users of PhishTank.com. They discuss whether the reported

[13] Blog post announcing retirement of legacy Google APIs: http://googlecode.blogspot.com/2010/11/introducing-google-apis-console-and-our.html

[14] Google documentation, *Search the entire web:* http://support.google.com/customsearch/answer/1210656?hl=en

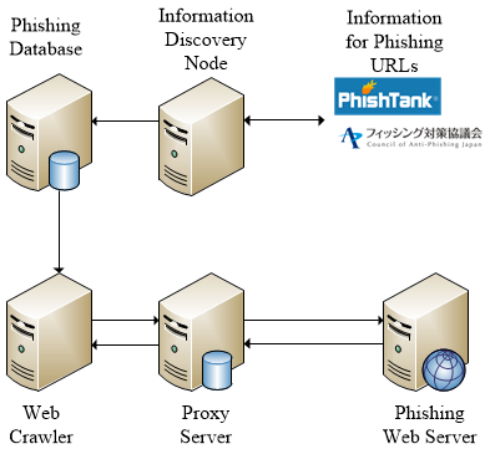


Figure 2.3: Crawler for phishing sites

sites are actually phishing attempts or not. According to an analysis report by Moore et al. [18], PhishTank submissions take approximately 46 hours on average to be verified.

Aside from the open resources, there is another data provider, the Anti-Phishing Working Group (APWG) [5]. The APWG has more than 3000 members from more than 1700 companies and agencies worldwide. The Council of Anti Phishing, Japan [8] operated by JP-CERT/CC also accepts the submission of emails that are considered to be phishing, and provides the URLs for the exclusive use of members.

Due to the nature of phishing, phishing emails also contain phishing URLs. Based on the text mining and heuristic-based analysis, it is possible to find new phishing URLs from phishing sites. For example, the University of Alabama (UAB) Phishing Team's data repository was used in past surveys for evaluating phishing blacklists. According to the survey, UAB has got in touch with several sources who share their spam as part of the UAB Spam Data Mine.

In order to develop an automated data collection system for phishing, our prototype implementation checks the URL of Phishing sites at external data sources as well as receives phishing URLs from data providers. There is a standard for the exchange format of phishing information, however it is not widely used. Instead, our prototype implementation extracts URLs from various sources, registers the URLs into a database, and then launches web crawlers for accessing the URLs.

We implemented a crawler to capture phishing contents. Based on phishing crawlers developed during past research work [25, 22], we crawled phishing sites by using a rendering engine present in modern browsers, to accommodate the presence of JavaScript. Note that phishing sites often employ JavaScript, whereas traditional web crawlers do not accommodate it.

Figure 2.3 illustrates our implemented system. First, an information discovery node receives the URL of phishing sites from external data sources, and it then stores the URL into a phishing database. A web crawler periodically checks the URL, and accesses the newly registered URL after adding a unique identifier for the website in the HTTP request header. The proxy server removes the identifier from the request and sends it to the phishing web server. After the proxy server receives web contents back from the server, it stores the contents related to the identifier. Such crawling sessions will finish when the whole website contents have been loaded or when an expiration time has been reached.

3 Dataset Description

This chapter describes the collected datasets, that belong to the network infrastructure layer and the end-point layer, coming from various data sources. The datasets allow us to observe threats in a wider scope, to compare observed threats in different institutions and regions, and to extract threat trends over a long-term period. Analysis modules, that are part of workpackage 2, process the data coming from these datasets and create knowledge based on the analysis outcome.

3.1 Statistics of Infrastructure Layer Datasets

This section presents the infrastructure-layer datasets highlighting key features and providing the dataset descriptions. The overview of the infrastructure layer datasets is presented below.

The datasets can be divided into five main categories:

3.1 Traffic

The traffic datasets include packet traces from an academic backbone network as well as from universities, Netflow and sFlow traces from Internet Exchange Points (IXes), an academic cloud system and universities. The packet traces allow detailed analysis at the packet level such as packet payload analysis and packet arrival timing analysis, while Netflow and sFlow traces are more readily available from routers and switches and allow analysis at the flow level.

3.2 DNS

The DNS datasets include DNS query and/or response logs from authoritative name servers including some deployed at universities, one root name server, as well as, cache resolver servers from universities. DNS is one of the key infrastructure systems, and provides the name

resolution, essential part of almost all communications over the Internet. The datasets allow us to analyze threats to the DNS systems. Also, many threats can be identified by signatures in hostnames so that the datasets can be used to find suspicious hostnames and their corresponding IP addresses actually used, which can be further analyzed using other datasets.

3.3 Topology

The topology datasets include topology information derived from OSPF and BGP routing protocols from an academic backbone. The routing is another essential part of the Internet infrastructure systems so that the datasets are used to identify threats to the routing systems as well as to find side effects of threats to the routing information.

3.4 Telescopes

The telescope datasets include traffic observed at two telescope systems, one in Japan and one in Europe. The telescope systems monitor a large unused address space which is supposed to have no normal traffic. Since all traffic to these addresses is suspicious, the datasets can be used to analyze possible network attacks such as source spoofed attacks and scanning activities.

3.5 Early Warning Systems

The Early Warning System (EWS) datasets include alarms produced by threat detection systems. The aim of EWS is to automatically detect new threats in their early stage, and provide alarms for further analysis. An EWS implements a systematic process to collect data, gather related information from third parties, and then analyze the collected information for particular threats of interest.

Therefore, the analysis engine of the EWS is part of workpackage 2, but the resulting alarms are included in workpackage 1 because the provided alarms can be inputs for further analysis.

The discrepancies in the description format of the datasets, between the infrastructure layer and the end-point layer, are caused by the fact, that infrastructure datasets and endpoint datasets might be described by different parameters. The format presented in this section unifies the description to some extent keeping fields relevant to the infrastructure datasets.

3.1.1 Traffic dataset

3.1.1.1 WIDE-TRANSIT packet traces with short payload

Description: Packet traces with short payloads collected from a 150 Megabit Ethernet link which connects WIDE and its upstream network. Data is taken everyday from 14:00 JST for 15 minutes.

Data Start Time: 14:00 JST (UTC+9:00) everyday since 2006-08-19

Data End Time: on going (as of 2016-03-14)

Data Duration: 15 minutes everyday

Data formats and database: pcap (gzipped)

Data size: Each file is about 2.5-8GB (gzipped) about 6.5TB in total (as of 2016-03-31)

Sampling Method (if applicable): 15 minutes everyday

Location of data collection: WIDE NOC at Otemachi, Tokyo, Japan

Contact Person: Kenjiro Cho (IIJ/WIDE)

Detailed Description: These traces consist of packets collected in both directions on a 150 Megabit Ethernet external link which connects WIDE backbone and its upstream network. The actual link capacity is 1Gbps with the capped bandwidth of 150Mbps [7].

The first 96 bytes including the Ethernet frame are captured for each packet. Data is captured everyday from 14:00 JST for 15 minutes. The traces are available in 15 minutes duration files in the pcap format containing both directions. NTP is used for clock synchronization.

Some flows are observed only in one direction due to asymmetric routing.

File naming convention: Each file is named by the start time (ccyy-mm-dd.gz) in local time (JST).

API for data access: N/A

Availability: The original datasets are available upon request.

An anonymized non-payload version of this dataset along with summary information is publicly available at the MAWI working group Traffic Archive web site[1] (under samplepoint-F).

[1] http://mawi.wide.ad.jp/mawi/

3.1.1.2 WIDE-TRANSIT aggregated flow data

Description: Aggregated flow data produced by a multi-dimensional flow aggregation tool, collected from a 150 Megabit Ethernet link which connects WIDE and its upstream network.

Data Start Time: 2013-02-07

Data End Time: ongoing (as of 2016-03-31)

Data Duration: Each file contains 5-minutes-long aggregated flow data.

Data formats and database: Custom (Aguri2 format)

Data size: Each file is about 50KB, about 17GB in total (as of 2016-03-31)

Sampling Method (if applicable): The flow data is aggregated every 30 seconds.

Location of data collection: WIDE NOC at Otemachi, Tokyo, Japan

Contact Person: Kenjiro Cho (IIJ/WIDE)

Detailed Description: The agurim tool [14] is used for multi-dimensional flow aggregation. The aggregated flow datasets are available in 5-minute-duration files in the Aguri2 text format. NTP is used for clock synchronization.

File naming convention: Each file is named by the start time (ccyy-mm-dd.HHMMSS.agr) in local time (JST).

API for data access: To be implemented. The JSON API for data query will be available soon.

Availability: The original datasets are available upon request.

An anonymized version of this dataset is publicly available at the MAWI working group Traffic Archive web site[2].

[2] http://mawi.wide.ad.jp/~agurim/

3.1.1.3 Packet Traces from a university

Description: Packet traces collected from a 10GbE link from a University Campus to WIDE.

Data Start Time: 2014-03-16

Data End Time: ongoing (as of 2016-03-14)

Data Duration: Each file is 15 minutes long. Dataset files are kept for 24 hours. Occasionally, traces are archived for further analysis.

Data formats and database: pcap.

Data size: The file size is about 2-8GB for each 15-minute-long trace file, 922GB in total.

Sampling Method (if applicable): N/A

Location of data collection: Kanagawa, Japan.

Contact Person: Kenjiro Cho (IIJ/WIDE)

Detailed Description: Packet traces collected from a 10GbE link of a university campus to WIDE. The university has 2 upstream ASes.

File naming convention: The trace files are kept for 24 hours in 96 rotating files. Each file is 15 minutes long. The files are rotated based on their names: log0, log1, ..., log95, from the newest to the oldest.

API for data access: N/A

Availability: The datasets are available upon request.

3.1.1.4 NetFlow data from universities

Description: NetFlow data exported from transit routers

Data Start Time: 2013-09-02

Data End Time: ongoing

Data Duration: 24 hours, 365 days a year

Data formats and database: nfdump format

Data size: 1.9TB

Sampling Method (if applicable): 1 out of 512 to 8192 samples, sampled by NetFlow

Location of data collection: Japan

Contact Person: Yuji Sekiya (The Univ. of Tokyo)

Detailed Description: The dataset is exported from Cisco routers using NetFlow. This NetFlow dataset does not include all flows, but sampled NetFlow. The sampling rate is 1 packet on every 512 to 8192 packets.

File naming convention: nfcapd.YYYYMMDDHHMM (every 5 minutes).

API for data access: MATATABI and n6 API

Availability: The original datasets are currently available to the *NECOMA* JP members only only. The dataset is also converted into the Hadoop Hive format.

3.1.1.5 sFlow data from Internet backbone

Description: sFlow data exported from Internet backbones

Data Start Time: 2013-08-13

Data End Time: ongoing

Data Duration: 24 hours, 365 days a year

Data formats and database: Text format from sflowtool

Data size: 1.87TB

Sampling Method (if applicable): 1 of 8192 samples, sampled by sFlow

Location of data collection: Japan

Contact Person: Yuji Sekiya (The Univ. of Tokyo)

Detailed Description: The datasets are exported from switches on Internet bacobone by sFlow.

File naming convention: sflow-IXNAME-YYYYMMDD.txt.gz

API for data access: MATATABI and n6 API

Availability: The original datasets are available to the *NECOMA* JP members only. The dataset is also converted into Hadoop Hive format.

3.1.1.6 sFlow data from a public cloud

Description: sFlow data exported from an IaaS Cloud, which is deployed in WIDE Project

Data Start Time: 2013-08-20

Data End Time: 2016-03-31

Data Duration: 24 hours, 365 days a year

Data formats and database: Text format from sflowtool

Data size: 138GB

Sampling Method (if applicable): 1 out of 8192 samples, sampled by sFlow.

Location of data collection: Japan

Contact Person: Yuji Sekiya (The Univ. of Tokyo)

Detailed Description: The datasets include traffic to/from IaaS cloud which accommodates over 350 VMs. The users of this IaaS cloud are WIDE Project members, including universities and research organization people.

File naming convention: sflow-cloud-net3-YYYYMMDD.txt.gz

API for data access: MATATABI and n6 API

Availability: The original datasets are currently available to the *NECOMA* JP members only. The dataset is also converted into Hadoop Hive format.

3.1.1.7 sFlow data from universities

Description: sFlow data exported from academic network

Data Start Time: 2013-08-20

Data End Time: 2016-03-31

Data Duration: 24 hours, 365 days a year

Data formats and database: Text format from sflowtool

Data size: 944GB

Sampling Method (if applicable): 1 out of 8192 samples, sampled by sFlow.

Location of data collection: Japan

Contact Person: Yuji Sekiya (The Univ. of Tokyo)

Detailed Description: sFlow data exported from universities, which are connected to WIDE Project

File naming convention: sflow-widebb-f6ote-YYYYMMDD.txt.gz

API for data access: MATATABI and n6 API

Availability: The original datasets are currently available to the *NECOMA* JP members only. The dataset is also converted into Hadoop Hive format.

3.1.2 DNS dataset

3.1.2.1 DNS query data from a WIDE DNS server

Description: pcap data of DNS queries on an authoritative DNS server in WIDE Project

Data Start Time: 2013-10-10

Data End Time: 2016-03-31

Data Duration: 24 hours, 365 days a year

Data formats and database: pcap

Data size: 398GB

Sampling Method (if applicable): dnscapture (libpcap)

Location of data collection: WIDE Project, Japan

Contact Person: Yuji Sekiya (The Univ. of Tokyo)

Detailed Deription: The dataset consists of pcap files captured on the DNS server in WIDE Project. The DNS server has both the roles of cache and authoritative servers. The pcap files only include packets filtered based on UDP and TCP port 53. This DNS server has several zones related to the domain name “wide.ad.jp” and several inverse zones.

File naming convention: dump-YYYYMMDDHHMM.gz

API for data access: MATATABI, and n6 API

Availability: The original datasets are available to the *NECOMA* JP members only. The dataset also supports the Presto-db query engine.

3.1.2.2 DNS query logs from cache resolver DNS servers in universities

Description: BIND query logs from cache resolver DNS servers in an universities

Data Start Time: 2013-07-24

Data End Time: 2016-03-31

Data Duration: 24 hours, 365 days a year

Data formats and database: Text data exported by BIND

Data size: 450GB

Sampling Method (if applicable): BIND querylog function

Location of data collection: Japan

Contact Person: Yuji Sekiya (The Univ. of Tokyo)

Detailed Description: The dataset is collected on a cache DNS server. The server software is ISC BIND and it has a feature of logging DNS queries. The dataset is collected by the feature.

File naming convention: YYYYMMDD/named-query.log.XXX (XXX is a number)

API for data access: MATATABI, and n6 API

Availability: The original datasets are available to the *NECOMA* JP members only. The dataset also supports the Presto-db query engine.

3.1.2.3 DNS query data from cache resolver DNS servers in universities

Description: pcap data captured on cache resolver DNS servers in universities

Data Start Time: 2013-10-04

Data End Time: 2016-03-31

Data Duration: 24 hours, 365 days a year

Data formats and database: pcap files

Data size: 3.3TB

Sampling Method (if applicable): dnscapture (libpcap)

Location of data collection: Japan

Contact Person: Yuji Sekiya (The Univ. of Tokyo)

Detailed Description: The dataset is collected on cache resolver DNS servers.

File naming convention: dump-YYYYMMDDHHMM.gz

API for data access: MATATABI, and n6 API

Availability: The original datasets are currently available to the *NECOMA* JP members only. The dataset also supports the Presto-db query engine.

3.1.2.4 DNS Query logs from authoritateive DNS servers in universities

Description: BIND query logs from authoritative DNS servers in universities

Data Start Time: 2013-09-23

Data End Time: 2016-03-31

Data Duration: 24 hours, 365 days a year

Data formats and database: Text data exported by BIND

Data size: 245GB

Sampling Method (if applicable): BIND querylog function

Location of data collection: The University of Tokyo, Tokyo, Japan

Contact Person: Yuji Sekiya (The Univ. of Tokyo)

Detailed Description: The dataset is collected on authoritative DNS servers. The DNS servers have over 200 zones.

File naming convention: YYYYMMDD/named-query.log.XXX (XXX is a number)

API for data access: MATATABI, and n6 API

Availability: The original datasets are currently available to the *NECOMA* JP members only. The dataset also supports the Presto-db query engine.

3.1.2.5 DNS query data from authoritateive DNS servers in universities

Description: pcap data captured at authoritative DNS servers in universities

Data Start Time: 2013-10-07

Data End Time: 2016-03-31

Data Duration: 24 hours, 365 days a year

Data formats and database: pcap files

Data size: 403GB

Sampling Method (if applicable): dnscapture (libpcap)

Location of data collection: The University of Tokyo, Tokyo, Japan

Contact Person: Yuji Sekiya (The Univ. of Tokyo)

Detailed Description: The dataset is collected on authoritative DNS servers. The DNS servers have over 200 zones.

File naming convention: dump-YYYYMMDDHHMM.gz

API for data access: MATATABI, and n6 API

Availability: The original datasets are currently available to the *NECOMA* JP members only. The dataset also supports the Presto-db query engine.

3.1.2.6 DNS query data from M-root DNS servers in DITL

Description: Full DNS traffic capture from all instances of M-Root DNS server.

Data Start Time: Varies for each DITL event

Data End Time: Varies for each DITL event

Data Duration: 50 hours (1 hour each prior/after the core 48 hours)

Data formats and database: Gzipped pcap files. Each file corresponds to 1 hour of traffic including DNS queries and responses.

Data size: 890GB in total for 2015 DITL event (50 hours). Note that each "packet" occupies about 90 bytes in average in the gzipped pcap file.

Sampling Method (if applicable): N/A

Location of data collection: Same as M-DNS traffic dataset. Note that the Osaka site has recently been added and no corresponding DITL data is available.

Contact Person: Akira Kato (Keio Univ.)

Detailed Description: DITL[3] is an annual synchronized traffic measurement effort initially coordinated by CAIDA since 2006. M-Root has been participating in DITL since 2007. In DITL, it is suggested to capture 50 hours full of traffic, and many Root DNS servers as well as other DNS servers have been participating. Most of the data has been uploaded to the DNS-OARC site[4]. This 9 DITL datasets (2007 through 2015) from M-Root are uploaded to one of the *NECOMA* servers available to the project partners. A brief summary is shown in Table 3.1.

Note that traffic data captured at AS112 server in Osaka (see below) is also available.

Also, other, smaller scale, synchronized traffic measurements were carried out. Thus, only DNS Queries were captured, and duration varies. All data were captured in 2010: Jan 13 (2hrs), Jan 20 (44hrs), Jan 26 (50hrs), Feb 9 (50hrs), Mar 2 (62hrs), Mar 23 (56hrs), May 4 (52hrs), May 25 (50hrs), Jul 14 (118hrs).

File naming convention: `DITL`*YYYY*`/`*instance*`/M-`*instance*`-`*YYYYMMDDHHmm*`.gz`

API for data access: ssh, account issued upon request

Availability: *NECOMA* partners only

[3] `http://www.caida.org/projects/ditl/`
[4] `http://www.dns-oarc.net/`

3.1.3 Topology dataset

3.1.3.1 OSPF topology datasets from the WIDE backbone

Description: OSPF topology snapshots of the WIDE backbone area collected every 2 hours.

Data Start Time: 2013-09-13

Data End Time: ongoing (as of 2016-03-31)

Data Duration: N/A

Data formats and database: Each dataset is a text file in Solana's custom format which looks like the following:

```
router <AREAID> <IPADDRESS> or
router <AREAID> <IPADDRESS> dns <DNSNAME: no spaces>
network <IPADDRESS> <MASK> <ROUTER ID ADDRESS LIST>
link broadcast <AREAID> <SRC_RTR> <OUTIF_IP> <DSTNETWORKIP>
link ptpt <AREAID> <SRC_RTR> <OUTIF_IP> <DST_RTR> <OUTIF_IP>
interface <AREAID> <IPADDRESS> <OUTIF_IP> <OUTIF_MASK> <TYPE> \
          <TOS> <METRIC> <CONFIGURED_BW> <UTILIZED_BW>
```

Data size: Each file is about 35KB, 366MB in total.

Sampling Method (if applicable): Snapshots are taken every 2 hours.

Location of data collection: University of Tokyo, Nezu, Tokyo.

Contact Person: Kenjiro Cho (IIJ/WIDE)

Detailed Description: OSPF topology snapshots of the WIDE backbone are collected by an appliance, SMARTHawk from Solana Networks. http://www.solananetworks.com/products/smart_hawk.html Snapshots are taken every 2 hours.

File naming convention: Each file is named after the timestamp (ccyymmdd_HHMMSS.stf) in local time (JST).

API for data access: Accessible by an HTTP GET request.

Availability: The datasets are available upon request..

3.1.3.2 iBGP datasets from WIDE (AS2500)

Description: This dataset contains BGP update messages and periodic RIB dumps collected from WIDE (AS2500). The quagga[5] software routing suite is used to collect and store the datasets.

Data Start Time: 2014-06-09

Data End Time: On going (as of 2016-03-31)

Data Duration: Continuous.

Data formats and database: The MRT format.

Data size: Each RIB snapshot compressed file is about 5MB, 40GB in total.

Sampling Method (if applicable): Snapshots are taken every 2 hours.

Location of data collection: The University of Tokyo, Tokyo, Japan

Contact Person: Kenjiro Cho (IIJ/WIDE)

Detailed Description: This dataset contains BGP update messages and periodic RIB dumps collected from WIDE (AS2500). The quagga software routing suite is used to collect and store the datasets. The routing information is collected through the feed from a route-reflector of WIDE.

File naming convention: ribs.ccyymmdd.HHMM.bz2

API for data access: The data files are downloadable from the web server

Availability: The datasets are available upon request.

[5] `http://www.quagga.net/`

3.1.4 Telescope dataset

3.1.4.1 Darknet traffic traces from NII

Description: Packet traces (some packets with short payload bytes) collected at a /18 network. Data is gathered 24 hours every day.

Data Start Time: 2006-08-25 11:25 JST (UTC+9:00)

Data End Time: ongoing

Data Duration: 24 hours every day

Data formats and database: pcap (gzipped)

Data size: Each file is between 50-500MB (gzipped). Three major data loss time periods: from 2007-05-27 to 2007-06-28, from 2010-11-26 to 2011-02-04, from 2012-01-09 to 2012-09-25. About 0.5TB in total (gzipped; as of 2016-03-30)

Sampling Method (if applicable): N/A

Location of data collection: a /18 subnetwork in Japan.

Contact Person: Kensuke Fukuda (NII)

Detailed Description: Collection of telescope traffic destined to one /18 allocated but unused IPv4 darknet address block in Japan since 2006 with three major data loss time periods: May 27th, 2007 to Jun. 28th, 2007; Nov. 26th, 2010 to Feb. 4th, 2011; Jan. 9th, 2012 to Sep. 25th, 2012. The complete packet headers (layer-2, -3, and -4) were captured and short payload bytes (just for some packets) into pcap format files are provided daily for the full 24 hours.

File naming convention: Each file is named after the start time (JST): packet.ccyy.mm.dd.hh.mm.ss.gz

API for data access: To be implemented.

Availability: Currently for NII internal use only

3.1.5 Early warning dataset

3.1.5.1 NASK: Port scans detected by ARAKIS

Description: detected port scans

Data start time: 2013-01-01

Data end time: ongoing

Data Duration: N/A

Data formats and database: according to the n6 platform specification

Data size: 5k events per day on average, see figure 3.1 for details

Sampling method (if applicable): N/A

Location of data collection: Poland

Availability: complete dataset available to consortium members only

API for data access: n6 API

Detailed Description: ARAKIS is an early warning system intended to improve situational awareness with regard to threats observed in the Polish address space. Its main data source is a large-scale distributed honeypot network. This dataset contains horizontal port scans (port sweeps) which were detected automatically by the system through analysis of connection attempts observed in the entire monitored address space. Each port scan is represented as a single event in the n6 API and contains the following properties:

- detection time
- transport layer protocol
- source port and IP address (with associated ASN and country code)
- destination port
- number of observed probes

3.1.5.2 NASK: Attacks detected by ARAKIS

Description: connections to honeypots containing suspicious payload

Data start time: 2013-01-01

Data end time: ongoing

Data Duration: N/A

Data formats and database: according to the n6 platform specification

Data size: 124k events per day on average, see figure 3.2 for details

Sampling method (if applicable): N/A

Location of data collection: Poland

Availability: complete dataset available to consortium members only

API for data access: n6 API

Detailed description: Most of the information collected by ARAKIS comes from a distributed network of server honeypots. Every connection to a honeypot is matched against a set of IDS rules corresponding to known suspicious payloads. This dataset contains information about connections that triggered at least a single rule. An attack is represented as a single event in the n6 API and contains the following properties:

- detection time
- transport layer protocol
- source port and IP address (with associated ASN and country code)
- destination port and anonymized IP address
- name of triggered rules

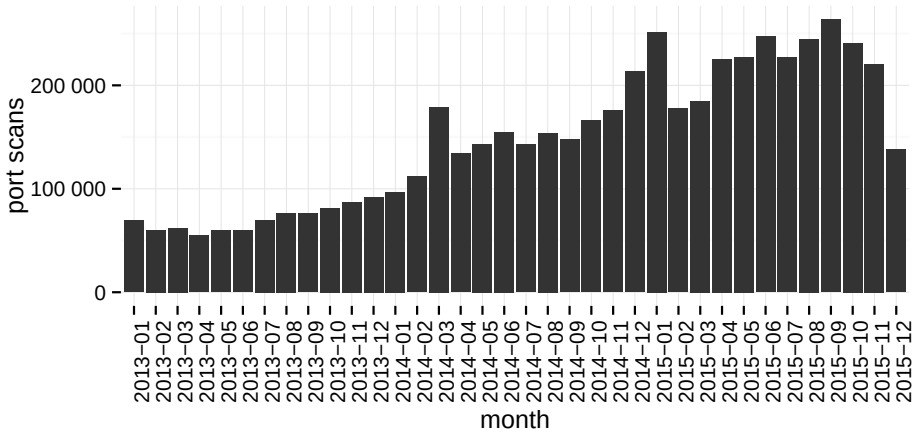


Figure 3.1: Monthly distribution of port scans detected by ARAKIS.

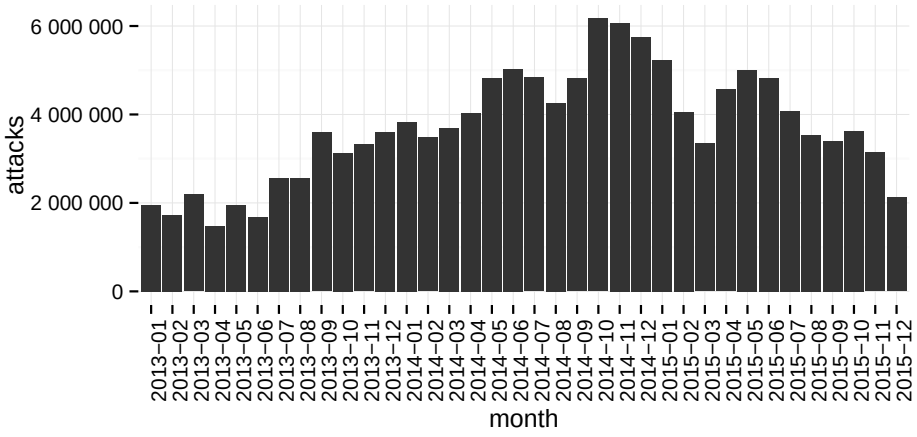


Figure 3.2: Monthly distribution of attacks detected by ARAKIS honeypots.

3.1.5.3 NEMU malware data

Description: The data are attacks incidents detected by NEMU through network traffic monitoring.

Data Start Time: 2013-01-01

Data End Time: ongoing

Data formats and database: log files that are parsed and stored in an MySQL database

Sampling Method (if applicable): N/A

Location of data collection: Greece

Contact Person: Thanasis Petsas (FORTH)

Detailed Description: NEMU is a shellcode detector based on network-level emulation. NEMU inspects the client-initiated data of each network flow, which may contain malicious requests towards vulnerable services. Any server-initiated data, such as the content served by a web server, are ignored. For TCP packets, the application-level stream is reconstructed using TCP stream reassembly. An IA-32 CPU emulator repeats the execution multiple times, starting from each and every position of the stream. NEMU scans the traffic towards any service and does not rely on exploit or vulnerability specific signatures, thus it is capable to detect polymorphic attacks to even less widely used or "forgotten" services.

Collected Data

For each identified attack, NEMU generates

- an alert with generic attack information and the execution trace of the shellcode
- a raw dump of the reassembled TCP stream
- a full payload trace of all attack traffic (both directions) in libpcap format
- the raw contents of the modified locations in the virtual memory of the emulator, i.e., the decrypted shellcode.

Deployment

The NEMU detector is deployed on a passive monitoring sensor that inspects the traffic of the access link that connects part of an educational network with hundreds of hosts to the Internet.

File naming convention: N/A

API for data access: n6 API

Availability: Currently available internally for FORTH only.

3.1.5.4 BotHunter botnet data

Description: Infection profiles produced by BotHunter network monitor.

Data Start Time: 2013-01-01

Data End Time: ongoing

Data formats and database: log files that are parsed and stored in a MySQL database

Sampling Method (if applicable): N/A

Location of data collection: Greece

Contact Person: Thanasis Petsas (FORTH)

Detailed Deription: BotHunter [6] is a network defense algorithm designed to detect whether a system is running coordination-centric malware (such as botnets, spambots, spyware, Trojan exfiltrators, worms, adware).

BotHunter monitors the two-way communication flows between hosts within your internal network and the Internet. It aggressively classifies data exchanges that cross your network boundary as potential dialog steps in the life cycle of an ongoing malware infection. BotHunter employs Snort as a dialog event generation engine, and Snort is heavily modified and customized to conduct this dialog classification process.

Dialog events are then fed directly into a separate dialog correlation engine, where BotHunter maps each host's dialog production patterns against an abstract malware infection lifecycle model. When enough evidence is acquired to declare a host infected, BotHunter produces an infection profile to summarize all evidence it has gathered regarding the infection. In short, BotHunter helps you rapidly identify infected machines inside your network that are clearly and helplessly under the control of external malicious hackers.

Deployment

BotHunter is currently deployed and monitors an educational network with hundreds of hosts to the Internet. Furthermore, a web-based graphical interface has been developed in order to facilitate the search of the collected information related with the infection incidents and the bot-like behavior of the monitored hosts.

File naming convention: N/A

[6] http://www.bothunter.net

API for data access: n6 API

Availability: Currently available internally for FORTH only.

3.1.5.5 DNS reflection attack data

Description: The data are analysis results of DNS reflection attacks based on sampled traffic and DNS query log data.

Data Start Time: depends on traffic data

Data End Time: depends on traffic data

Data Duration: depends on traffic data

Data formats and database: Hive query log

Data size: 100KB per day

Sampling Method (if applicable): 1 out of 8192 sampled by sFlow and NetFlow

Location of data collection: depends on traffic data

Contact Person: Kazuya Okada (NAIST)

Detailed Description: A measurement in December 2013, based on sFlow and DNS queries, was an early indication of a DDoS campaign against specified organizations. More specifically, we analyzed traffic volume based on sFlow datasets and DNS queries on a DNS openresolver which observed multiple networks. If a large DNS reflection attack is initiated, we can observe changes to the traffic volume.

File naming convention: {flow type}_{flow source}_YYYYMMDD.dat

API for data access: Implementation planned

Availability: Internally to NAIST only

3.1.5.6 NTP reflection attack data

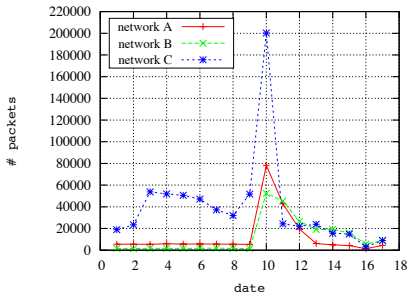


Figure 3.3: Daily NTP packet volumes in Feb. 2014

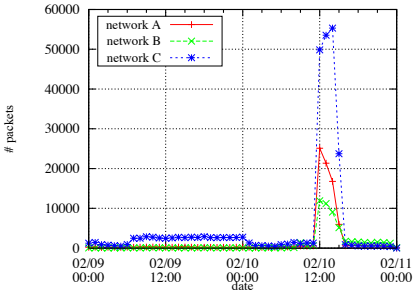


Figure 3.4: Hourly NTP packet volumes between Feb.9th and Feb.11th, 2014

Description: The data are analysis results of NTP reflection attacks based on sampled traffic data.

Data Start Time: depends on traffic data

Data End Time: depends on traffic data

Data Duration: depends on traffic data

Data formats and database: use traffic data, which collected by sFlow and NetFlow

Data size: 10KB per day

Sampling Method (if applicable): 1 out of 8192 sampled by sFlow and NetFlow

Location of data collection: The University of Tokyo, Tokyo, Japan

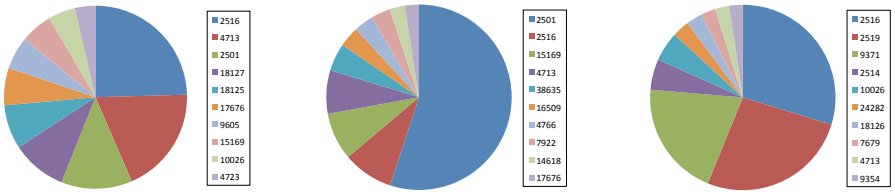
Contact Person: Kazuya Okada (NAIST)

Detailed Description: We have been observing a growth in the number of reflective DDoS attacks using NTP (Network Time Protocol) recently. Most notably, the most recent reflective DDoS campaign was reported to reach 400Gbps at its peak, on February 11th, 2014.

Through our ongoing measurement of NTP, we were able to identify an early indication of the campaign on February 10th, 2014. The threat dataset includes sFlow traffic data which is collected at the backbone network of WIDE project. We analyzed changes in the volume of NTP traffic against internal hosts which do not publicly provide NTP service.

Figure 3.3 shows daily NTP packet volumes which were collected by sFlow in three networks. We could find volume changes around February 10th 2014. To reveal more details of the phenomenon, Figure 3.4 shows hourly NTP packet volumes between February 9th and February 11th. The changes started at 12:00 on February 10th. Actually, we observed DDoS attacks against routers in our network using NTP packets on February 10th.

We converted the source IP address of each attack packet to an AS number which distributes the IP address based on IRR data base. Figure 3.5 shows top 10 AS number in each network. Most of the listed AS numbers are from domestic ISPs.



[1] Network A [1] Network B [1] Network C

Figure 3.5: Top10 NTP packet source list in Feb.10th 2014.

File naming convention: {flow type}_{flow source}_YYYYMMDD.dat

API for data access: Implementation planned

Availability: Internally to NAIST only

3.1.6 Summary

We have been collecting a wide variety of datasets from various network infrastructures, including internal networks and the Internet. The datasets

were inputs to workpackage 2, the cooperative, cross-layer analysis on multiple datasets.

Most of the datasets are continuously collected, thus they will evolve over time, even after the termination of the project.

But most importantly, given the diversity of the datasets, when combined, the datasets complemented each other in the analysis phase, generating valuable threat information that enabled the development of new defense mechanisms covering the infrastructure layer.

3.2 Statistics of Endpoint Layer Datasets

We use the term "endpoint layer" to refer to all kinds of security information that is collected from the perspective of consumer devices (e.g., personal computers, smartphones) or users. These kind of datasets are application-specific, which allows us to analyze threats for particular applications in depth.

Datasets available to the consortium can be divided into the following categories: *mail and messaging*, which focus on spam; *web-related*, including phishing and SSL server response data; *user behavior*, in particular related to the assessment of phishing websites' credibility; *sinkholes* for registering bot activity. Finally, *client honeypots and sandboxes* provide information that may be used to identify malicious websites, monitor botnet communication and detect other types of hostile activity.

Most of the datasets are shared through the common API described in Section 2.1.3.

The discrepancies in the description format of the datasets, between the infrastructure layer and the end-point layer, are caused by the fact, that infrastructure datasets and endpoint datasets might be described by different parameters. The format presented in this section unifies the description to some extent keeping fields relevant to the endpoint datasets.

3.2.1 Mail and messaging dataset

3.2.1.1 KEIO: spam

Short description: spam mails sent to `kato@wide.ad.jp`.

Data start time: 2008-01-01

Data end time: 2015-03-31

Storage: raw RFC821 [7]-compliant format

[7] RFC821: http://tools.ietf.org/html/rfc821

Approximate data size: 850MiB∼3.12GiB per year, 136k∼508k messages per year

Location of data collection: Japan

Contact person: Akira Kato (Keio Univ.)

Availability: consortium members only

Detailed description: In order to deliver an email to the specified email address, IronPort, an Email Security Appliance currently from Cisco Systems, investigate the message and adds a "X-Spam: Yes" line in the header if IronPort considers it as a spam.

File naming convention:

`kato/`*YYYY*`/`*sequential-number*

Note that a missing number represents a message that IronPort considered as a ham – not all of these messages were actually hams, however. Note that some of the messages classified as spam could be hams. Please do not disclose spam donors' privacy.

Short description: spam mails sent to *username*`@sfc.wide.ad.jp`.

Data start time: 2015-04-01

Data end time: 2016-03-31

Storage: raw RFC821 [8]-compliant format

Approximate data size: 536MiB∼1.05GiB per month, 33.0k∼50.5k messages per month

Location of data collection: Japan

Contact person: Akira Kato (Keio Univ.)

Availability: consortium members only

Detailed description: In order to deliver an email to the specified email address, IronPort, an Email Security Appliance currently from Cisco Systems, investigate the message and adds a "X-Spam: Yes" line in the header if IronPort considers it as a spam. This is a superset (while timing is different) of SPAM messages sent to a perticular person.

File naming convention:

`SPAM/`*YYYY*`/`*MMDD*`/`*sequential-number*

[8] RFC821: http://tools.ietf.org/html/rfc821

3.2.1.2 UTokyo: spam

Short description: spam mails sent to `sekiya@wide.ad.jp` and `sekiya@nc.u-tokyo.ac.jp`

Data start time: 2012-05-31

Data end time: 2016-03-31

Storage: raw RFC821-compliant format

Approximate data size: 290MiB∼500MiB per year, 26k∼40k messages per year

Location of data collection: Japan

Contact person: Yuji Sekiya (The Univ. of Tokyo)

Availability: consortium members only

Detailed description: The spam dataset includes the mails delivered to sekiya@wide.ad.jp and sekiya@nc.u-tokyo.ac.jp and detected as spam by spamassassin software on the mail server. The fields "X-Spam-Score:" and "X-Spam-Flag:", are included in each mail and show the degree of spam. Note that some of the messages classified as spam could be hams. Please do not disclose spam donors' privacy.

3.2.2 Web dataset

3.2.2.1 UT: phishing URLs

Short description: phishing URL, IP address, AS number of the phishing server, and WHOIS information

Data start time: 2012-01-04

Data end time: ongoing (as of 2016-03-31)

Storage: PostgreSQL [9], Hive [10],

Approximate data size: 1,950,037 records

Location of data collection: The University of Tokyo, Japan

Contact person: Daisuke Miyamoto (The Univ. of Tokyo)

Availability: consortium members only

Detailed Description: This dataset primarily consists of URLs of phishing websites, as well as IP addresses, AS numbers, domain names, WHOIS lookup results. Phishing URLs were provided by PhishTank[11] and the Council of Anti Phishing, Japan (CAPJ)[12]. PhishTank is a reporting site for phishing URLs, and it is the biggest data provider worldwide. CAPJ provides newly found phishing sites' URLs via emails. The sites are verified as phishing by the CAPJ technical staff.

In order to collect phishing URLs, our crawler checks `phishtank.com` every 15 minutes. It also extracts URLs from CAPJ emails and consequently checks these URL. The crawler also stores the collected information into a database.

This database is structured as shown in Table 3.2, where the sub ID field is the ID used by the particular data provider. For example, if the phishing URL is provided as `http://www.phishtank.com/phish_detail.php?phish_id=1234567`, the number 1234567 is stored as the sub ID in our database. The verified field is a flag indicating whether the site is really a phishing one or not. Verification of a reported URL is done by voting of the registered users for PhishTank. In the case of CAPJ, the verification process is done by CAPJ operators.

[9] PostgreSQL: `http://www.postgresql.org/`
[10] Hive: `https://hive.apache.org/`
[11] PhishTank: `http://www.phishtank.com/`
[12] CAPJ: `https://www.antiphishing.jp/enterprise/url.html` (in Japanese)

3.2.2.2 UT: phishing content

Short description: phishing websites including HTML files, image files and script files

Data start time: 2013-06-01

Data end time: ongoing (as of 2016-03-31)

Storage: PostgreSQL, Hive, and media files

Approximate data size: 341 GiB for phishing content, 263 GiB for screen shots

Location of data collection: The University of Tokyo, Japan

Contact person: Daisuke Miyamoto (The Univ. of Tokyo)

Availability: consortium members only

Detailed Description: This dataset stores the content of phishing websites. Based on phishing crawlers developed in past research work [25, 22], we crawled phishing sites using the rendering engine of a modern browser supporting JavaScript. Note that Phishing sites often employ JavaScript, whereas traditional web crawlers do not support it. A web crawler periodically checks the URLs in the phishing URL dataset, and accesses the newly registered URLs. Upon access, it also adds a unique identifier in the HTTP request header. The proxy server removes the identifier from the request, and then sends it to a phishing web server. After the proxy server receives the web contents from the server, it stores the content related to the identifier. Such crawling sessions will finish when the whole contents have been loaded from the website or when threshold time has been reached.

This database is structured as shown in Table 3.3, where the index points at file names of the downloaded contents. Text strings are extracted by accessing `document.text` elements in the browser.

3.2.2.3 IMT: SSL server response

Short description: response payload from stimulated SSL servers

Data start time: 2010-07-01

Data end time: 2014-03-25

Storage: binary files (to be read using Parsifal software[16]), can be extracted to database format (MongoDB [13])

Approximate data size: 170+ GiB (for 12 campaigns, ranging from 2.5 GiB to 22 GiB – in average 14.5 GiB)

Location of data collection: Télécom SudParis, France

Contact person: Gregory Blanc (IMT)

Availability: an n6 interface has been set up at `http://phoenix.telecom-sudparis.eu:2534/ssl_n6/`. Access is granted to `anonymous:anonymous` at the moment of writing. Available commands are: `ssl.json`, `certificate.json`, `certificate_status.json` and `server_status.json`. More details on the syntax is available in Deliverable D3.3.

Detailed description: SSL/TLS measurement campaigns were launched in July 2010 and 2011 to assess the quality of HTTPS servers [17]. The provided datasets comprise these two periods as well as some other measurement campaigns from which results are publicly available and carried out by other independent institutions such as the Electronic Frontier Foundation [12]. The collection campaigns performed at Télécom SudParis were based on the active enumeration of open HTTPS ports (TCP/443) over the entire IPv4 space. The July 2010 campaign only initiated SSL communication with services responded on port 443 through full TCP handshake initiated by a single `ClientHello` TLS message. On the contrary, the July 2011 campaigns feature several `ClientHello` messages containing different protocol versions, ciphersuites and TLS extensions.

Table 3.4 shows contents present in the collected responses. The results may differ with the campaign (identified by a 3-digit number) and a time period (month and year of collection) for each probed server. Fields followed by an asterisk (*) indicate possibly missing fields when the server has not properly answered with a handshake.

[13] MongoDB: `https://www.mongodb.org/`

3.2.3 User behavior dataset

Short description: behavior of users while assessing credibility of websites

Data start time: 2013-12-13

Data end time: 2015-03-31

Storage: CSV [14], PostgreSQL, Hive, and media files

Approximate data size: 100 kiB for users' decision, 9.6 GiB for eye-tracking video

Location of data collection: The University of Tokyo, Japan

Contact person: Daisuke Miyamoto (The University of Tokyo)

Availability: consortium members only

Detailed description: This dataset contains two types of data: one is concerned with decision results and criterion collected by means of questionnaire, while the other is concerned with eye-movement records collected by an eye-tracking camera.

It is structured as shown in Table 3.5, where the decade is defined as follows: it has value of 1 if participants are aged under twenty, 2 for participants in their twenties, 3 if in their thirties, 4 if in their forties, and 5 for participants aged 50 and older. The decision field is the decision result, 1 for labelling the site as legitimate, and 2 for phishing. The criterion field contains the reasons why a participant labelled a site as legitimate or phishing. The following options were presented to the participant: "Content of Web page," "URL of the site," "Security Information of Browser," and "Other Reason." The participants then proceded to select all reasons applicable to their decision (multiple answers allowed) and described further details if they selected "Other Reason" option.

Our experimental setup process is described below. It must be noted that our experiments must not collect and/or analyze personally identifiable information. The experimental design, concept and methodologies for recruiting participants are also explained below.

1. Recruiting participants by online and poster advertising at a college campus.
2. Explaining our experiment to the participant.

[14] CSV: http://en.wikipedia.org/wiki/Comma-separated_values

- Our purpose is to observe users' activity, particularly how they assess the credibility of websites.
- Our goal is to develop security mechanisms for protecting users from phishing.
- Before the experiments, each participant will be asked his/her age and sex.
- During the experiments, each participant will be monitored by an eye-tracking device, and be shown 20 websites. Their activity will be monitored, and they will be asked if each website seems to be phishing or not.
- Collected data consists of participants' age, sex, decision result, decision criterion, and eye-tracking data.
- Collected data is shared with both European and Japanese members of the consortium.
- Collected data will be shared with third-parties, which research and develop security technologies.

3. Showing 20 website screenshots, including legitimate websites and pseudo phishing sites.
   In the experiment, the phishing sites are not real phishing sites, in order to avoid information leakage. Instead, our participants saw 20 screenshots of a browser that rendered the websites. These screenshots were taken on Windows 7 equipped with IE 10.0.
4. Asking "how do you assess the credibility of this website?" for 20 websites.
5. Paying remuneration.

As shown in Table 3.6, we prepared 12 phishing sites and eight legitimate ones for the test. By comparison, a typical phishing IQ test [11] presented participants with 13 phishing sites and seven legitimate ones, so the ratio of phishing sites is not very different.

The participants who were likely to check "URL of the site" would fail to flag websites 5, 14 and 17, since these sites had almost the same URL as the legitimate sites except for one letter. The URLs of the websites 7, 12, and 19 contained a legitimate-sounding domain name. Website 20 was legitimate but the domain name of this site had no indication of its brand names. For participants who tended to check "Security Information of Browsers", websites 11 and 20 might be difficult to assess because they were phishing sites but presented participants with valid SSL certificates. Conversely, websites 6 and 9 were legitimate but did not employ valid SSL certificates though they required users to login. Of course, our prepared phishing websites

were lookalikes of the legitimate ones. It might have been more difficult for the participants who relied on "Content of Web page."

The eye tracking data is composed of two types of files. One is a video file in AVI format that captured eye position and eye movements on the screen, and the other one is a CSV file containing records of eye-movement, namely time, eye position on the screen, and category of eye movements. John et al. [15] classified the eye movements into four categories, namely Saccades, Fixations, Smooth pursuit movements, and Vestibulo-ocular reflexes. Research in experimental psychology has evidenced a strong link between eye movements and mental disorders [9, 19]. Generally, the saccadic eye movement changes with what a person is seeing. In the context of mental model, Irwin et al. showed that the mental rotation is suppressed during the movements [13], and Tokuda [24] showed that mental workload, the indicator of how mentally/cognitively busy a person is, can be estimated from saccadic intrusions.

3.2.4 Sinkhole dataset

3.2.4.1 NASK: data from sinkholes

Description: connections to domains sinkholed by CERT Polska

Data start time: 2014-01-01

Data end time: ongoing

Data Duration: N/A

Data formats and database: according to the n6 platform specification

Data size: 672k events per day on average, see figure 3.6 for details

Sampling method (if applicable): N/A

Location of data collection: Poland

Availability: complete dataset available to consortium members only

API for data access: n6 API

Detailed description: CERT Polska sinkholes multiple malware-related domains by redirecting traffic to a server under its control. Information about connecting bots is continuously fed into the n6 platform for sharing and analysis. In the n6 API, a single event corresponds to one received connection or a group of connections. Multiple connections from a single source address to the same IP and port are grouped as a single event if time difference between events is small (threshold depends on the current configuration, usually it is less than an hour). Event attributes:

- time of the first connection
- time of the last connection in a group (only if grouped)
- count of all connections in a group (only if grouped)
- bot IP address and port
- destination (sinkhole) address and port
- bot family

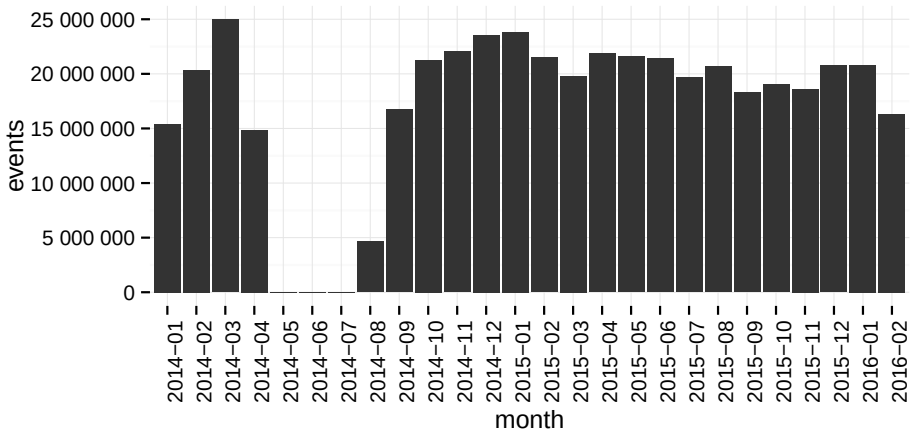


Figure 3.6: Monthly distribution of bot sightings on the sinkhole run by CERT Polska. Note: historic data for April–August 2014 was not imported into the current version of the n6 platform.

3.2.5 Client honeypots and sandbox dataset

3.2.5.1 NASK: network connection attempts by malware

Short description: outgoing connections from all executables analyzed automatically in sandboxes operated by CERT Polska

Data start time: 2014-02-21

Data end time: 2015-02-02

Storage: n6 platform

Data size: 450 URLs per day on average, see figure 3.7 for details

Location of data collection: Poland

Availability: complete dataset available to consortium members only

Detailed description: Many of suspicious binary samples obtained by CERT Polska are analyzed automatically in a sandbox. From reports generated by the sandbox software, information about network communication is extracted and provided through the n6 platform. Analysis of a sample can generate any number of events corresponding to network activity. These events may contain:

- protocol, destination IP address, source and destination ports
- domains and URLs if available
- occurrence count if multiple connections with the same destination address and port were observed

The dataset may contain communication between bots and C&C servers but a significant part of connection attempts will be with benign servers. This is a consequence of the fact that not all analyzed samples are malicious, and even if they were, malware often connects to benign services in order to verify its internet connectivity or IP address.

3.2.5.2 NASK: peer-to-peer bot list

Short description: data obtained from P2P botnet crawlers created by CERT Polska

Data start time: 2013-12-03

Data end time: 2015-04-07

Storage: n6 platform

Data size: after 2 initial months, 3k events per day on average, see figure 3.8 for details

Location of data collection: Poland

Availability: complete dataset available to consortium members only

Detailed description: CERT Polska created crawlers that use reverse-engineered P2P botnet protocol to observe communication occurring within such networks. By employing such crawlers, it is possible to discover the majority of infected machines in a botnet. Discovery of a bot is represented as a single event in the n6 API, with the following details:

- bot IP address, protocol and port used for communication
- botnet name

Sightings of a single bot that occur in a short time interval are grouped into a single event.

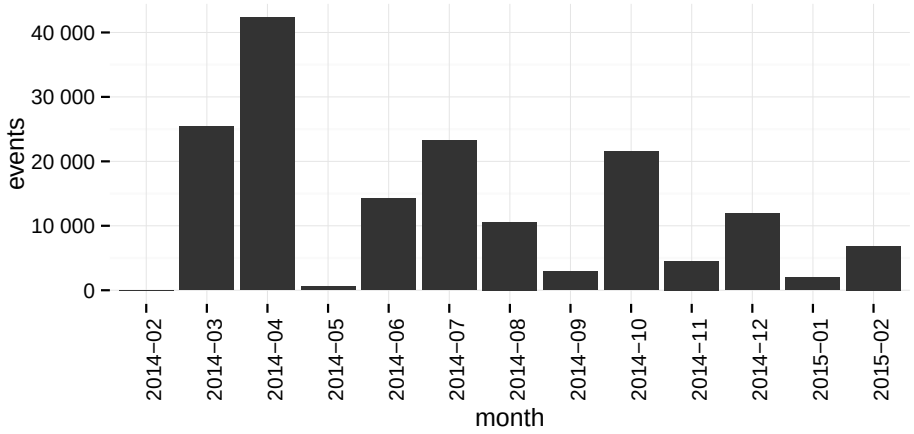


Figure 3.7: Monthly distribution URLs observed in the sandbox environment.

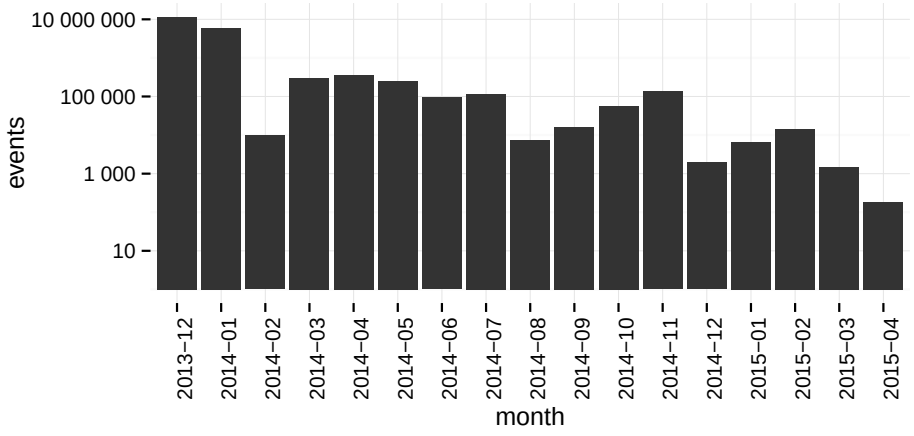


Figure 3.8: Monthly distribution of peer-to-peer bot sigtings (logarithmic scale).

3.2.6 Third-party dataset

3.2.6.1 NASK: malicious URLs collected from multiple sources

Description: malicious URLs reported by other sources

Data start time: 2013-01-01

Data end time: ongoing

Data Duration: N/A

Data formats and database: according to the n6 platform specification

Data size: 78k events per day on average, see figure 3.9 for details

Sampling method (if applicable): N/A

Location of data collection: N/A

Availability: used internally

API for data access: n6 API

Detailed Description: Information about malicious URLs provided by multiple third-parties are collected by the n6 platform for operational and research purposes. Most of the sources cannot be disclosed. The biggest (in terms of volume of data) public source integrated with the platform is VirusWatch [15].

[15] VirusWatch Watching adress changes of Malware Url's: http://lists.clean-mx.com/cgi-bin/mailman/listinfo/viruswatch

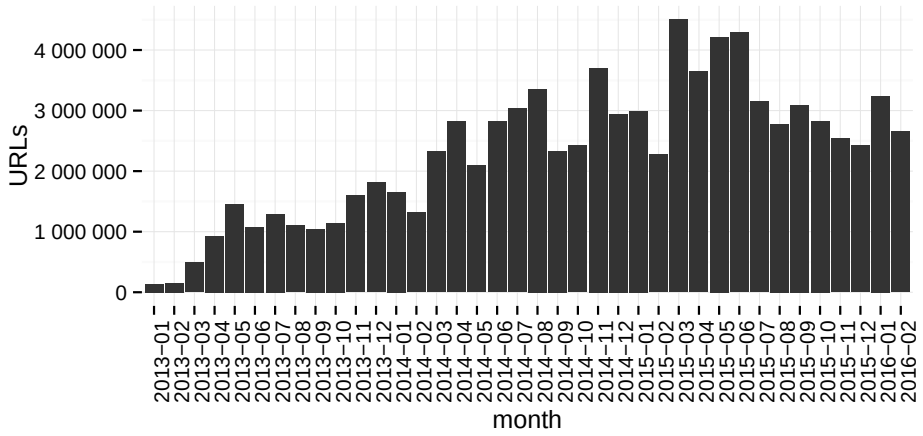


Figure 3.9: Monthly distribution of reports of malicious URLs collected by the n6 platform.

3.2.7 Summary

There exists a great variety of information that can be relevant to endpoint layer security and this fact is reflected in the diversity of datasets contributed by members of the *NECOMA* consortium. With the exception of Mail and Messaging datasets, where different datasets contain the same kind of data, all datasets listed in this document are focused on different types of information and were created using various collection methods.

Obtaining interesting endpoint layer data may be more difficult compared to infrastructure-level information, since it often requires active engagement in network communication with observed entities. Steps necessary to crawl the Web or other overlay networks present a good example of the complexity of endpoint layer data collection – one must establish a connection to a remote server, request resources using appropriate application-level protocol, parse the response, and choose next actions (e.g., following links), which depend on the received content. In contrast, many kinds of infrastructure data can be obtained through passive monitoring, leveraging logging capabilities offered by existing hardware and software.

We believe that the challenges associated with acquisition and interpretation of endpoint layer data are the main reason why its systematic collection is less common than corresponding efforts done on the infrastructure-level. This difference is reflected in the datasets available to the consortium since a majority were classified into the “infrastructure” category.

The importance of endpoint layer data should not be underestimated, as they provide high-level information, often related to interaction with users.

3.3 Availability

Many of the datasets are available internally only to the consortium members because of the sensitive nature of security related datasets, but most datasets can be partially shared to the broader community by request. In addition, some of the datasets are made publicly available. Our datasets will be a valuable contribution to the cybersecurity research community because most researchers do not have direct access to these types of datasets.

Table 3.1: DITL events to which M-Root has participated

Year	Start	End	Size	Note
2007	Jan 9 0000 UTC	Jan 11 0000 UTC	240GB	
2008	Mar 18 0000 UTC	Mar 20 0000 UTC	268GB	
2009	Mar 30 0000 UTC	Apr 2 0000 UTC	466GB	74hrs
2010	Apr 13 1400 UTC	Apr 15 2300 UTC	466GB	61hrs
2011	Apr 12 1200 UTC	Apr 14 1200 UTC	503GB	
2012	Apr 17 1200 UTC	Apr 19 1200 UTC	577GB	
2013	May 28 1200 UTC	May 30 1200 UTC	567GB	
2014	Apr 15 1200 UTC	Apr 17 1200 UTC	697GB	
2015	Apr 13 1200 UTC	Apr 15 1200 UTC	890GB	

Table 3.2: phishing URL dataset

field	type	description
ID	int	identification number
vendor	int	ID for information source
sub ID	int	ID used in vendors
verified	int	boolean flag for the URL is verified
crawled	int	boolean flag for crawled or not
URL	text	phishing site's URL
FQDN	text	FQDN retrieved from URL
Domain name	text	domain name retrieved from URL
IPv4 address 1	text	lookup result(A) with our DNS resolver
IPv4 address 2	text	lookup result(A) with public DNS resolver
IPv6 address 1	text	lookup result(AAAA) with our DNS resolver
IPv6 address 2	text	lookup result(AAAA) with public DNS resolver
whois	text	WHOIS result for domain name
AS number	text	aslookup result of IPv4 (1) address

Table 3.3: phishing content dataset

field	type	description
ID	int	identification number
index	text	file lists of content
HTTP code	int	HTTP response code
text	text	text extracted by the web browser
screenshot	binary	screen shot captured by the web browser

Table 3.4: SSL server response dataset

field	format
server IP address	IPv4
answer type	$empty\|handshake\|alert\|junk$
protocol version*	$sslv2\|sslv3\|TLSv1.0\|TLSv1.1\|TLSv1.2$
ciphersuite(s)*	one or several supported ciphersuites identifiers
RSA key*	size in bits (modulo 8)
certificate start*	date
certification expiration*	date
certificate issuer*	X.500 directory information

Table 3.5: User decision dataset

field	type	description
ID	int	ID for the participant
decade	int	definition of age by decade
sex	int	boolean flag for sex, male or female
decision	int	boolean flag for decision, phishing or not
criterion	int	criterion while assessing website's credibility

Table 3.6: Conditions of each site used for the participant-based test

#	Website	Phish	Lang	Description
1	Google	no	JP	SSL
2	Amazon	yes	JP	`tigratami.com.br`, once reported as a compromised host
3	Sumishin Net Bank	no	JP	EV SSL
4	Yahoo	yes	JP	`kazuki-j.com`, once reported as a compromised host
5	Square Enix	yes	JP	`secure.square-enlix.com`, similar to legitimate URL `secure.square-enix.com`
6	Ameba	no	JP	non-SSL
7	Tokyo Mitsubishi UFJ Bank	yes	JP	`bk.mufg.jp.iki.cn.com`, similar to legitimate URL `bk.mufg.jp`
8	All Nippon Airways	yes	JP	IP address
9	Gree	no	JP	non-SSL
10	eBay	no	EN	EV SSL
11	Japan Post Holdings	yes	JP	`direct.yucho.org`, SSL
12	Apple	yes	EN	`aaple.com.uk.sign.in...`
13	DMM	no	JP	SSL
14	Twitter	yes	JP	`twittelr.com`
15	Facebook	yes	JP	IP address
16	Rakuten Bank	yes	JP	`vrsimulations.com`, once reported as a compromised host
17	Sumitomo Mitsui Card	yes	JP	`www.smcb-card.com`, SSL
18	Jetstar Airways	no	JP	SSL, non pad-lock icon by accessing non-SSL content
19	PayPal	yes	EN	`paypal.com.0.security-c...`
20	Tokyo-Tomin Bank	no	JP	3rd party URL `www2.answer.or.jp`, EV SSL

4

Datasets in Research

This chapter describes the outcomes and achievements to which the *NECOMA* datasets contributed to. It includes eleven academic papers and one article.

4.1 Academic Papers

Title Random Projection and Multiscale Wavelet Leader Based Anomaly Detection and Address Identification in Internet Traffic

Authors Romain Fontugne, Patrice Abry, Kensuke Fukuda, Pierre Borgnat, Johan Mazel, Herwig Wendt, and Darryl Veitch

Publish Proceedings of the 40th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Brisbane, Australia, Apr.19-24, 2015

Dataset WIDE-TRANSIT Packet Traces with short payload in the Traffic Dataset

Abstract We present a new anomaly detector for data traffic, 'SMS', based on combining random projections (sketches) with multiscale analysis, which has low computational complexity. The sketches allow 'normal' traffic to be automatically and robustly extracted, and anomalies detected, without the need for training data. The multiscale analysis extracts statistical descriptors, using wavelet leader tools developed recently for multifractal analysis, without any need for timescales to be selected a priori. The proposed detector is illustrated using a large recent dataset of Internet backbone traffic from the MAWI archive, and compared against existing detectors.

Title An Empirical Mixture Model for Large-Scale RTT Measurements

Authors Romain Fontugne, Johan Mazel, and Kensuke Fukuda

Publish Proceedings of the 34th IEEE International Conference on Computer Communications (INFOCOM 2015), Hong Kong, Apr.26-May.1, 2015

Dataset WIDE-TRANSIT Packet Traces with short payload in the Traffic Dataset

Abstract Monitoring delays in the Internet is essential to understand the network condition and ensure the good functioning of time-sensitive applications. Large-scale measurements of round-trip time (RTT) are promising data sources to gain better insights into Internet-wide delays. However, the lack of efficient methodology to model RTTs prevents researchers from leveraging the value of these datasets. In this work, we propose a log-normal mixture model to identify, characterize, and monitor spatial and temporal dynamics of RTTs. This data-driven approach provides a coarse grained view of numerous RTTs in the form of a graph, thus, it enables efficient and systematic analysis of Internet-wide measurements. Using this model, we analyze more than 13 years of RTTs from about 12 millions unique IP addresses in passively measured backbone traffic traces. We evaluate the proposed method by comparison with external data sets, and present examples where the proposed model highlights interesting delay fluctuations due to route changes or congestion. We also introduce an application based on the proposed model to identify hosts deviating from their typical RTTs fluctuations, and we envision various applications for this empirical model.

Title Identifying Coordination of Network Scans Using Probed Address Structure

Authors Johan Mazel, Romain Fontugne, and Kensuke Fukuda

Publish Proceedings of International Workshop on Traffic Measurement and Analysis (TMA), Louvain La Neuve, Belgium, Apr. 7-8, 2016

Dataset WIDE-TRANSIT Packet Traces with short payload in the Traffic Dataset

Abstract A great deal of work has been devoted to the study and detection of scanning. Existing detection of isolated probing, however, only provides an incomplete picture of scanning activities. Coordinated probing using several hosts, in particular, cannot be accounted for with simple scan detection that expects a single source. In this paper, we apply run length encoding concepts to characterize IP address structure of scanning events. We then employ graph techniques to uncover hidden coordinated network scans as communities. These coordinated events are split according to destination port and targeted network prefixes. We evaluate the sensitivity of our method with synthetic data and verify that our method outperforms the state-of-the-art for both stub and backbone network monitoring. Finally, we provide a detailed analysis of several coordinated scans occurring in real network traffic. Using these results, we verify that our method is reliable and extracts coordinated scans that are very consistent in terms of network traffic characteristics.

Title Characterizing Roles and Spatio-Temporal Relations of C&C Servers in Large-Scale Networks

Authors Romain Fontugne, Johan Mazel, and Kensuke Fukuda

Publish Proceedings of International Workshop on Traffic Measurements for Cybersecurity (WTMC), Xian, China, May 30, 2016

Dataset All datasets in the Traffic Dataset

Abstract Botnets are accountable for numerous cybersecurity threats. A lot of efforts have been dedicated to botnet intelligence, but botnets versatility and rapid adaptation make them particularly difficult to outwit. Prompt countermeasures require effective tools to monitor the evolution of botnets. Therefore, in this paper we analyze 5 months of traffic from different botnet families, and propose an unsupervised clustering technique to identify the different roles assigned to C&C servers. This technique allows us to classify servers with similar behavior and effectively identify bots contacting several servers. We also present a temporal analysis method that uncovers synchronously activated servers. Our results characterize 6 C&C server roles that are common to various botnet families. In the monitored traffic we found that servers are usually involved in a specific role, and we observed a significant number of C&C servers scanning the Internet.

Title Classification of SSL Servers based on their SSL Handshake for Automated Security Assessment

Authors Sirikarn Pukkawanna, Youki Kadobayashi, Gregory Blanc, Joaquin Garcia-Alfaro, and Hervé Debar

Publish Proceedings of the 3rd International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS), Wroclaw, Poland, Sep. 11, 2014

Dataset SSL dataset in the Web Dataset

Abstract The Secure Socket Layer (SSL) and Transport Layer Security (TLS) are the most widely deployed security protocols used in systems required to secure information such as online banking. In this paper, we propose three handshake-information-based methods for classifying SSL/TLS servers in terms of security: (1) Distinguished Names-based, (2) protocol version and encryption algorithm-based, and (3) combined vulnerability score-based methods. We also classified real-world SSL/TLS servers, active during July 2010 to May 2011, using the proposed methods. Finally, we propose 45 features, deemed relevant to security assessment, for future SSL/TLS data collection. The classification results showed that servers had bimodal distribution, with mostly good and bad levels of security. The results also showed that the majority of the SSL/TLS servers had seemingly risky certificates, and used both risky protocol versions and encryption algorithms.

Title AJNA: Anti-Phishing JS-based Visual Analysis, to Mitigate Users' Excessive Trust in SSL/TLS

Authors Pernelle Mensah, Gregory Blanc, Kazuya Okada, Daisuke Miyamoto, and Youki Kadobayashi

Publish Proceedings of the 4th International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS), Kyoto, Japan, Nov. 5, 2015

Dataset SSL dataset in the Web Dataset

Abstract HTTPS websites are often considered safe by the users, due to the use of the SSL/TLS protocol. As a consequence phishing web pages delivered via this protocol benefit from that higher level of trust as well. In this paper, we assessed the relevance of heuristics such as the certificate information, the SSL/TLS protocol version and ciphersuite chosen by the servers, in the identification of phishing websites. We concluded that they were not discriminant enough, due to the close profiles of phishing and legitimate sites. Moreover, considering phishing pages hosted on cloud service platform or hacked domains, we identified that the users could easily be fooled by the certificate presented, since it would belong to the rightful owner of the website. Hence, we further examined HTTPS phishing websites hosted on hacked domains, in order to propose a detection method based on their visual identities. Indeed, the presence of a parasitic page on a domain is a disruption to the overall visual coherence of the original site. By designing an intelligent perception system responsible for extracting and comparing these divergent renderings, we were able to spot phishing pages with an accuracy of 87% to 92%.

Title EyeBit: Eye-Tracking Approach for Enforcing Phishing Prevention Habits

Authors Daisuke Miyamoto, Takuji Iimura, Gregory Blanc, Hajime Tazaki, and Youki Kadobayashi

Publish Proceedings of the 3rd International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS), Wroclaw, Poland, Sep. 11, 2014

Dataset The User Behaviour Dataset

Abstract This paper proposes a cognitive method with the goal to get end users into the habit of checking the address bar of the web browser. Earlier surveys of end user behavior emphasized that users become victims to phishing due to the lack of knowledge about the structure of URLs, domain names, and security information. Therefore, there exist many approaches to improve the knowledge of end users. However, the knowledge gained will not be applied unless end users are aware of the importance and develop a habit to check the browser's address bar for the URL structure and relevant security information. We assume that the habit of checking the bar will improve educational effect, user awareness of secure information, and detection accuracy even in the case of sophisticated phishing attacks. To assess this assumption, this paper conducts a participant-based experiment where 23 participants' eye movement records are analyzed, and observes that novices do not tend to have the said habit. We then consider a way for them to acquire these habits, and develop a system which requires them to look at the address bar before entering some information into web input forms. Our prototype named EyeBit is developed as a browser extension, which interacts with an eye-tracking device to check if the user looks at the browser's address bar. The system deactivates all input forms of the websites, and reactivates them only if the user has looked at the bar. This paper shows the preliminary results of our participant-based experiments, and discusses the effectiveness of our proposal, while considering the potential inconvenience caused by EyeBit.

Title Eye Can Tell: On the Correlation between Eye Movement and Phishing Identification

Authors Daisuke Miyamoto, Gregory Blanc, and Youki Kadobayashi

Publish Proceedings of the 22nd International Conference on Neural Information Processing of the Asia-Pacific Neural Network Assembly (ICONIP), Istanbul, Turkish, Nov. 10, 2015

Dataset The User Behaviour Dataset

Abstract It is often said that the eyes are the windows to the soul. If that is true, then it may also be inferred that looking at web users' eye movements could potentially reflect what they are actually thinking when they view websites. In this paper, we conduct a set of experiments to analyze whether user intention in relation to assessing the credibility of a website can be extracted from eye movements. In our within-subject experiments, the participants determined whether twenty websites seemed to be phishing websites or not. We captured their eye movements and tried to extract intention from the number and duration of eye fixations. Our results demonstrated the possibility to estimate a web user's intention when making a trust decision, solely based on the user's eye movement analysis.

Title FP-tree and SVN for Malicious Web Campaign Detection

Authors Michał Kruczkowski, Ewa Niewiadomska-Szynkiewicz, and Adam Kozakiewicz

Publish Proceedings of the 7th Asian Conference Intelligent Information and Database Systems (ACIIDS 2015), Lecture Notes in Computer Science vol. 9012, 193-201, Bali, Indonesia, March 2015

Dataset The Web Dataset and the DNS Dataset

Abstract The classification of the massive amount of malicious software variants into families is a challenging problem faced by the network community. In this paper we introduce a hybrid technique combining a frequent pattern mining and a classification technique to detect malicious campaigns. A novel approach to prepare malicious datasets containing URLs for training the supervised learning classification method is provided. We have investigated the performance of our system employing frequent pattern tree and Support Vector Machine on the real malware database consisting of data taken from numerous devices located in many organizations and serviced by CERT Polska. The results of extensive experiments show the effectiveness and efficiency of our approach in detecting malicious web campaigns.

Title System for detection of malware campaigns (System do wykrywania kampanii złośliwego oprogramowania)

Author Michał Kruczkowski

Publish Telecommunications Review and Telecommunications News (Przeglad telekomunikacyjny i wiadomosci telekomunikacyjne), vol. 8-9, 789-797. September 2015 (in Polish)

Dataset The Web Dataset and the DNS Dataset

Abstract Detection of malicious campaigns is an extremely important issue, as it concerns a real need to provide security in the dynamically growing network. The MalCAS (Malware Campaign Analysis System) described in this paper uses machine learning and data exploration methods and fully addresses the needs of modern networks. The system's structure allows analysis of heterogeneous data from different layers of the network model.

Title Analysis of malicious campaigns in multiple heterogeneous threat data-sources

Author Michał Kruczkowski

Publish PhD thesis, 2015

Dataset The Web Dataset and the DNS Dataset

Abstract This doctoral dissertation concerns the problems of identification of malware (malicious software) campaigns on the Internet. This is an extremely important issue because it arises from a real need to ensure safety in rapidly growing computer networks. The proposed approach assumes the use of data mining and machine learning methods. It utilizes data about threat incidents taken from multiple data sources related with various layers of ISO/OSI network communication model. The results of the doctoral dissertation confirm that the automated analysis and classification of data from heterogeneous data sources can be efficiently used to protect the Internet. The use of full information about threats, including various network layers, leads to achieve better results compared with frequently used single layer analysis. Data mining and machine learning methods can efficiently support the cybercrime protection systems. The system MalCAS (Malware Campaign Analysis System) for malware campaigns identification that implements these methods fully corresponds the demands of contemporary networks. It can be applied as a useful and powerful tool to support the software environment ensuring network cybersecurity.

4.2 Articles

Title Flash Report of ShellShock Attacks (Japanese)

Author Yuji Sekiya

Publish http://www.necoma-project.jp/ja/blog/j992tt, Oct. 23, 2014

Dataset The Web Dataset and the DNS Dataset

Abstract An flash report describing the trend of ShellShock attacks on *NECOMA* Blog. The report used URL tracking dataset and pick up the characteristic URL patterns of the attacks.

5 Conclusion

The main goal of this document was to compile and present the outcomes of Tasks that were part of work package 1, but also results that were a consequence of work package 1 efforts.

This deliverable might be perceived as the final report of the design and implementation of the threat analysis platform going step by step through the process. Most of the content of this document was already presented in other deliverables throughout the NECOMA project, but due to the classified nature of the deliverables, the content was not disclosed until now.

This document outlines the process of investigation and establishment of a robust, but at the same time flexible, data sharing API and protocol, that were used within the NECOMA project to facilitate multilayer data correlation. We show not only the design, but also working implementations of technologies that were created during the course of the project that allow dataset analysis from a multi-layer perspective. And last but not least, we enumerate the enormous and diverse multi-layer set of datasets we were able to collect and make use of throughout the project.

This document demonstrates all the ingredients, from the bottom up, of a beyond-state-of-the-art threat analysis system that is a direct outcome of the NECOMA project.

Bibliography

[1] CybOX (Cyber Observable eXpression). http://cybox.mitre.org/. (Accessed 12th March 2014).

[2] MAEC (Malware Attribute Enumeration and Characterization). http://maec.mitre.org/. (Accessed 12th March 2014).

[3] n6 Platform. http://n6.cert.pl/. (Accessed 1st March 2014).

[4] STIX (Structured Threat Information eXpression). http://stix.mitre.org/. (Accessed 12th March 2014).

[5] Anti Phishing Working Group. APWG: Committed to Wiping Out Internet Scams and Fraud. Available at: http://www.apwg.com.

[6] T. Berners-Lee, R. Fielding, and L. Masinter. Uniform Resource Identifier (URI): Generic Syntax, 2005.

[7] K. Cho, K. Mitsuya, and A. Kato. Traffic data repository at the WIDE project. In *USENIX 2000 FREENIX Track*, San Diego, CA, June 2000.

[8] Council of Anti Phishing. Provide of Phishing sites' URL. Available at: http://www.antiphishing.jp/enterprise/url.html. (in Japanese).

[9] T. Crawford, S. Higham, T. Renvoize, J. Patel, M. Dale, A. Suriya, and S. Tetley. Inhibitory control of saccadic eye movements and cognitive impairment in alzheimer's disease. *Biol Psychiatry*, 9(57):1052–1060, 2005.

[10] R. Danyliw, J. Meijer, and Y. Demchenko. The Incident Object Description Exchange Format. RFC 5070 (Proposed Standard), 2007.

[11] R. Dhamija, J. D. Tygar, and M. A. Hearst. Why Phishing Works. In *Proceedings of Conference On Human Factors In Computing Systems*, Apr. 2006.

[12] Electronic Frontier Foundation. https://www.eff.org/observatory.

[13] D. E. Irwin and J. R. Brockmole. Mental rotation is suppressed during saccadic eye movements. *Psychonomic Bulletin and Review*, 7(4):654–661, 2000.

[14] M. Kato, K. Cho, M. Honda, and H. Tokuda. Monitoring the dynamics of network traffic by recursive multi-dimensional aggregation. In *OSDI2012 MAD Workshop*, Hollywood, CA, Oct. 2012.

[15] R. J. Leigh and D. S. Zee. *The Neurology of Eye Movements*. Oxford University Press, 4th edition, 1991.

[16] O. Levillain, H. Debar, and B. Morin. Parsifal: writing efficient and robust binary parsers, quickly. In *8th International Conference on Risks and Security of Internet and Systems*, page 1, La Rochelle, France.

[17] O. Levillain, A. Ébalard, B. Morin, and H. Debar. One year of ssl internet measurement. In *Proceedings of the 28th Annual Computer Security Applications Conference*, ACSAC '12, pages 11–20, New York, NY, USA, 2012. ACM.

[18] T. Moore and R. Clayton. Evaluating the Wisdom of Crowds in Assessing Phishing Websites. In *Proceedings of the 12th International Financial Cryptography and Data Security Conference*, Jan. 2008.

[19] B. Noris, K. Benmachiche, J. Meynet, J.-P. Thiran, and A. Billard. Analysis of Head-Mounted Wireless Camera Videos for Early Diagnosis of Autism. *Advances in Soft Computing*, 45:663–670, 2007.

[20] OpenDNS. PhishTank - Join the fight against phishing. Available at: http://www.phishtank.com.

[21] OpenDNS. Providing A Safer And Faster Internet. Available at: http://www.opendns.com.

[22] S. Sheng, B. Wardman, G. Warner, L. F. Cranor, J. Hong, and C. Zhang. An Empirical Analysis of Phishing Blacklists. In *Proceedings of the 6th Conference on Email and Anti-Spam*, Jul. 2009.

[23] H. Tazaki, K. Okada, Y. Sekiya, and Y. Kadobayashi. MATATABI: Multi-layer Threat Analysis Platform with Hadoop. In *3rd International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security*, page 8, Sept. 2014.

[24] S. Tokuda, G. Obinata, E. Palmer, and A. Chaparro. Estimation of mental workload using saccadic eye movements in a free-viewing task. *Proceedings of the 33rd Annua International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 4523–4529, August 2011.

[25] Y. Zhang, J. Hong, and L. Cranor. CANTINA: A Content-Based Approach to Detect Phishing Web Sites. In *Proceedings of the 16th World Wide Web Conference*, May 2007.