HSL: a Cyber Security Research Facility for Sensitive Data Experiments

Frédéric Beck, Abdelkader Lahmadi, Jérôme François

HSL: a Cyber Security Research Facility for Sensitive Data Experiments

Frédéric Beck, Abdelkader Lahmadi, Jérôme François
Université de Lorraine, CNRS, Inria, Loria, F-54000 Nancy, France
Email: {firstname.lastname}@inria.fr

***Abstract*—In this paper, we detail the design of a cybersecurity facility to carry reproducible and long term research activities in a safe environment, including malware collection and analysis, network telescopes and honeypots, or hosting critical services, without worrying about side effects or loss of data. The facility, aka High Security Lab (HSL), is running since 2010, and is widely used by multiple research groups to carry sensitive data cybersecurity experiments. It includes an evolving infrastructure with tools and processes for building and running long-term and reproducible cyber security experiments. We report on our experience and lessons learned from the design, the setup and the evolution of this facility during 10 years while focusing on major cybersecurity experiments that have been conducted by researchers.**

***Index Terms*—reproducible experiments, cybersecurity, research facility**

I. Introduction

Providing efficient and reproducible cybersecurity experiments at the research level requires highly secure and collaborative facilities for data collection, storage and analysis tasks. Quite frequently, research groups make these tasks in adhoc tools, with little support and maintenance which may lead their data to be lost and may effect severely the reproduction of the results, and the guarantee of continuous research activities. As a result of the lack of mature procedure and facilities, the approach taken in practice has usually been to carry experiments with different analysis tools, without common agreement, and to store the data in heterogeneous format, which makes their sharing within a research group or with others a difficult and complicated task, which limits collaborations, and leads to duplicating efforts. Although this approach allows few number of researchers to carry cybersecurity experiments, it is not possible to extend it to wider communities for long term collaborative actions. Most successful experiments are only localized, using specific-purpose testbed, and depend on actions of individuals as opposed to resulting from systematic processes in a controlled facility.

A large body of literature exists regarding the design and the development of cyber security testbeds for research and education. Mirkovic and al [1] described the DeterLab which is an open experimental facility, based on Emulab technology, allowing users to safely test security threats and defenses, including compromising hosts, or overloading and crashing them. A testbed for cyber physical systems security experiments is described in [2]. This testbed is mainly tailored to cyber defense competitions, allowing participants to test their defensive strategies with real-time feedback using scenarios including both cyber and cyber-physical elements. A more special-purpose testbed, dedicated to Microgrid cyber security is described in [3]. The testbed integrated multiple level of simulation, hardware-in-the-loop, and virtualisation to allow appropriate level of fidelity in attacks measurement and defense against the physical system. More testbeds and facilities are also developed by networking communities in particular for network measurement. A major project in this area is CAIDA [4], which provides a research infrastructure and a collaborative environment for large-scale data collection, curation and sharing. The project operates active and passive measurement infrastructures to provide data for Internet mapping and performance measurement at different locations, and also multiple software tools for monitoring Internet security and stability.

978-3-903176-32-4

Compared to existing testbeds, our research facility described in this paper is not designed to support a specific purpose testbed or deploying active or passive monitoring probes at different locations. However, it is well suited to support a broad class of cybersecurity experiments, by providing to researchers tools and a safe environment to carry their experiments, without compromising the security of the handled data, and also without introducing any risk on the hosting institution. Multiple experiments are carried by researchers on our facility, including malware collection and analysis, network telescopes for monitoring Internet background radiation, honeypots for collecting attack traces, hosting sensitive systems just to name a few. The main contributions of this paper are as follows:

- We detail the design and the technical features of our research facility for sensitive data experiments.
- We describe major experiments that are carried on our facility, while focusing on its benefits for running them.
- We report on the lessons learned from researcher experience, participants and resources perspectives.

The rest of the paper is organized as follows. In section II, we describe the design and the architecture of the HSL facility while providing technical details of its setup and deployment in our hosting institution. In section III, we detail major experiments on sensitive data that have been conducted through our facility. Finally, in section IV, we highlight the lessons learned while conducting and supporting such experiments.

II. Design and Implementation

The High Security Laboratory (HSL) is designed to act as a central hub for the technical aspects of cyber-security experiments where sensitive data is stored, and their analysis is performed without compromising their security. In the next sections, we present the architecture and objectives of the HSL infrastructure.

A. Overall Architecture

The HSL is designed to host sensitive data research activities in order to collect and store data while ensuring their confidentiality and integrity, both logically and physically. It is a cyber-security oriented datacenter and testbed to offer a safe and persistent environment for researchers to carry their experiments. Its overall architecture is shown in Fig. 1.

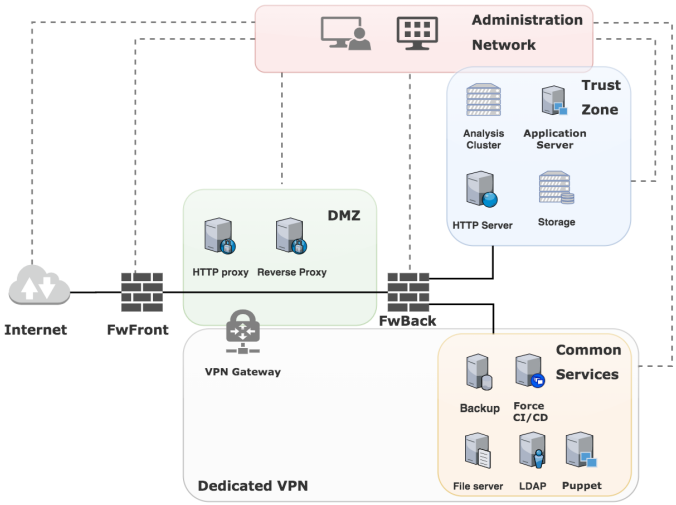

Fig. 1. HSL networking and hosting architecture.

The facility is composed of around 100 servers, organized in per-project clusters and isolated zones, deployed incrementally based on the needs of the hosted projects while anticipating future requirements and experiments. At the time of writing this paper, the HSL provides, overall, 3K cores (8 to 40 cores per server), 7TB+ memory (32 to 128 GB memory per server), 550 TB of disk space (1 to 20 TB disk space per server) including a 72TB backup storage unit. Different technical solutions are deployed based on the running projects or their usage, with the goal to maximize the utilisation of the hardware servers by relying on VmWare/Xen virtualization, ELK cluster, Docker containers, or dedicated servers for GPU computation. This architecture allows to address the 3 main roles and objectives of the infrastructure:

- realtime data collection and cybersecurity datasets provider for the research community, including historical data over several consecutive years;
- large scale and Internet wide experimentation;
- secure hosting and storage of sensitive data or services.

The platform is running with a limited administration staff, on average two part-time permanent persons, since our main focus is to guarantee confidentiality rather than high availability. Indeed, we rely on many automated tools, for servers and services deployment and configuration (Puppet), automatic backup (Bacula), constant monitoring (Icinga) and alerting via various channels (Mail, Mattermost chat via webhooks), using redundancy wherever it is required and possible. If this does not ensure 24/7 availability, it allows us to get back online quickly after a crash, usually a few minutes for a server, and between 30 minutes to 1 hour for the whole infrastructure.

B. Network Connectivity and Isolation

The HSL relies on trust zones, dedicated and isolated environments with limited and controlled interactions with the Internet. Each environment benefits from all the services offered by the HSL (network and data protection, automatic backup, local services package mirrors, DNS, LDAP, NTP) while being always separated from the outside world by two levels of security from different constructors/technologies (two firewalls from different constructors for the logical aspects, two different biometric authentication mechanisms for the physical ones), as shown in Fig. 1. Such trust zones are deployed for each hosted project, including its own network and VLAN to ensure that it is isolated from other hosted projects, but also its user accounts and groups in the HSL LDAP directory, its associated firewalling and users/groups access lists policies (ACLs). Access to such a trust-zone is possible via dedicated VPN networks, including dedicated gateways. These zones are fully integrated to the automatic configuration and software management solution (puppet) deployed in the infrastructure. The access to a trust zone is possible through a dedicated VPN, deployed exclusively for each project, and limited to the user accounts linked to the projects LDAP groups. The interconnection between and within these trust zones is ensured by a private 1Gb network.

Internet access is granted via various network connections. For daily usage, we have a dedicated /24 network (256 public IP address) with our own BGP routes and DNS zone registered globally, provided by a national provider for Education and Research communities. For specific experiments, and to have a point of comparison with a customer grade contract, we deployed also non-professional Fiber/ADSL connections. The most important requirement is to have a complete control over the security rules and policies enforced on these connections, so that we can have a complete control on our experiments. No direct access to or from the Internet is allowed in a trust zone. We limit the interactions with the outside world, by enforcing Internet access connections to go through proxies and reverse proxies, and thus limiting the attack surface. The only exception to this policy is the network telescope that we are hosting and presented in section III-A, which uses a dedicated subnet and a physical network in our front-end firewall. This whole /28 subnet is directly connected to the Internet with no firewall or any other security rules applied to the incoming traffic, and is white listed in our network provider's system to avoid false positive alerts. Outgoing traffic is monitored to detect compromission, but all the hosted

services are emulated, and the servers hosting these probes are hardened to avoid compromission.

Moreover, having our own network connections and BGP routes, independent from any other connections of our affiliated research institutions, and for which we are clearly identified as a source or destination of the traffic, is a mandatory feature to perform sensitive experiments. This means that if one of our experiment results in our IP range being blacklisted, or if we suffer from an attack in retaliation of a publication, for example a Distributed Denial of Service (DDoS) after the publication by a research group of an experiment for taking down a botnet [5], we can easily isolate the network and take the appropriate measures without impacting the hosting research center.

C. Data collection and storage

Due to the constraints in terms of physical security, we are focusing on data confidentiality and integrity and not high availability. Our main concern is to ensure that data are not leaked or lost/corrupted, whatever the situations (attack, power outage). Each server in the HSL embeds a RAID controller with its own battery to avoid filesystem corruption, i.e. all transactions end correctly. If a crash occurs, only the system disk is impacted. We are using puppet to automate tools and services installation and configuration. Crash recovery consists in reinstalling a base system and attaching it to puppet, which then ensures the server is back up quickly with minimal interaction and delay. When possible, we rely on data replication, in particular for the Elasticsearch-Logstash-Kibana (ELK) cluster that we use to store collected data (see section II-D), where each data index is separated in 2 shards and each shard is replicated twice.

To avoid data loss, an automatic backup solution based on Bacula has been deployed. On each server, placing data (or symbolic links) in a fixed directory will ensure the data will be backed up. We have tailored backup recipes for specific hosts and services, such as our DNS master, the Puppet server, and other tools deployed (see section). All raw data from our different probes and non reproductible datasets are backed up as well. Every day we perform incremental backups, differential backups every Friday, full backups on the first Friday of every month. Backup are kept for 365 days. These backups are stored on a dedicated storage unit, composed of twelve 8TB disks in RAID 5 with 2 hot spares, for a global available storage space of 72TB. At the moment, no external backup is performed, all data is backed up locally, as we have no other facility with the same security level available to replicate the data. Manual backup operations of critical data are performed on removable storage units or off-site servers, and rely on encryption to protect the data in these less secure environments. This task has to be performed by each hosted project and is highly recommended.

D. Tools

In order to limit as much as possible the interactions with the outside world, while allowing researchers to work on their data without compromising them outside the HSL, we deployed several tools to make sure they can perform data analysis within the HSL.

A complete CI/CD toolchain based on Gitlab has been deployed, allowing researchers to perform project planning and source code management. This allows data scientists and researchers to design, implement, and deploy all tools, software and services, including public services or dashboard for dissemination of their results, without exposing sensitive data to the outside world. A Mattermost messaging solution, allowing incoming webhooks, is deployed alongside this forge, and allows monitoring of the DevOps platform and tasks, and secure communication between team members.

By enforcing strict servers and services configuration with an automatic solution such as Puppet, we make sure everything is up and configured as it should. Many other services have been deployed to allow time synchronisation via NTP, packets installation via APT, authentication via LDAP... without interacting with the Internet. Accessing public software repositories such as Github or Python / Node modules is still possible via our proxies.

The monitoring and supervision of the whole HSL infrastructure and services is performed via Icinga which is a Nagios clone. Basic metrics and indicators are checked periodically on each server, and specific checks have been implemented and deployed for core services, and all automatic tasks. Any user or project can request their custom checks, which can then alert them via email or webhook.

As presented in section III-A, one of the main roles of the HSL is to collect realtime information and act as a data provider for researchers. Huge amounts of data are collected every day, and are enriched automatically before being indexed in the ELK cluster. This allows to quickly explore the available datasets, quickly generate basic statistics, and build visualization and dashboard that can be made publicly accessible.

III. Experiment Examples

Each experimentation and project hosted in the HSL has to receive the validation of an ethical committee, composed of legal, scientific and technical experts from our research instituions and national security agency. In this section, we describe major studies and projects carried by researchers and hosted in the HSL over the past years.

A. Cyber Threat Intelligence

Monitoring and predicting cyber-threats at the scale of Internet is a core research topic at HSL. Several types of security sensors are hosted:

- Honeypots have been deployed since 2008. Indeed, several versions and updates have been applied over years. Our solution relies on Modern Honey Network[1] to manage low-interaction honeypots variants from the state-of-the-art to mimic several types of services including SSH,

[1] https://github.com/pwnlandia/mhn

SQL, HTTP, SIP, etc. Nowadays, we collect around 100K hits per day, mostly on SSH, SQL and SIP honeypots.
- A network telescope, (darknet or Internet black hole) is a /20 network containing 4096 IP adresses to silently collect incoming traffic. This sensor has been started in 2014 with a daily rate of 4M events at that date while our current daily average is around 18M.
- Blacklists: we aggregate information about blacklisted IP addresses from several public sources such as DNS-BH[2], SSLBL[3], FireHOL[4], etc.

The darknet sensor is currently the main source of data to extract important knowledge about ongoing and future threats. However, most of data are very similar: 95% are TCP SYN packets (mostly SYN scans) and 50% target TCP port 23 (telnet) illustrating attackers looking for telnet accessible hosts such as embedded devices (e.g. IoT). Although these data sounds very usual and not revealing anything new, researchers demonstrated through several studies that relevant knowledge can be gathered. In [6], Topological Data Analysis (TDA) was leveraged by authors to cluster data (unsupervised machine learning). The main idea was to aggregate microscopic events (incoming packets) into large events (a DDoS attack, a syn scan, etc) to make more understandable data. They show that this technique outperforms the Suricata IDS (Intrusion Detection System) to label our data. Labelling collected data is actually a challenging but necessary tasks when operating a darknet in order to extract insights and trends about ongoing attacks in Internet. However, correlating darknet with blacklists is not satisfactory because less than one percent of IP addresses can be identified.

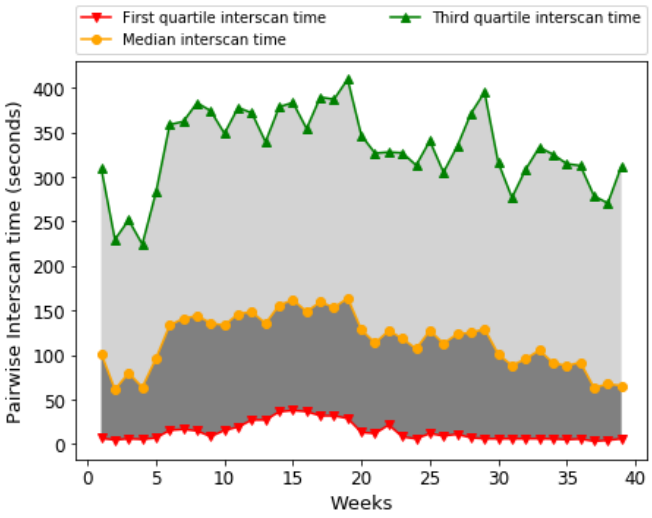

Fig. 2. Scans observed by the darknet, January - September 2015

Looking at large scan campaigns reveals new trends in attack targets and strategies. In [7], the authors modeled sequence of scans in a behavioral graph and highlighted that some specific environments, such as medical environment, were targeted. The authors in [8] observe smarter scans as shown in Figure 2. In this figure the inter-probing time by a scanner (identified by its IP address) is reported and, as seen, most of scans are quite slow and stealthy. In addition, the authors derive a semantic similarity between TCP port numbers that can be integrated into any algorithm where TCP ports must be compared. For example, this metric has been used to proactively block TCP scans in another network. Therefore, this result shows that knowledge inferred from darknet data can be successfully transferred into another domain where an analysis is performed. Although giving full and public access to darknet data is not possible, the researchers decided to make the inferred metric, and its monthly and and weekly updates, publicly available [5].

B. Malware collection and analysis

The highly secured infrastructure of the HSL allows to safely experiment well known cybercriminals techniques. Indeed, researchers have been able to reproduce in-lab a whole attack chain, including the phases of victim identification, vulnerabilities discovery, the implementation of infection tools and finally the trap that initiates the attack. Technically, they operate a server specialized in the choice of the exploit to be delivered to the victim, a malicious web server that redirects requests to the exploit kit server, and finally several vulnerable workstations that model the victims' machines. These targets are attacked through the victim's browser with a simple click by exploiting a dropper. This highlights the limitations of defense tools which, despite commercial announcements, are still unable to deal with this kind of threat, namely fileless attacks, where everything happens in memory.

Another aspect concerns the collection of malware for scientific analysis, in order to understand the techniques used by the attackers so that researchers can define new defense mechanisms. Since 2010, we hare able to collect over the years a considerable amount of malicious samples, consisting of 35 million files to date. This diversity of malicious code shows the increasing sophistication of malware diversion or obfuscation techniques. To cope with them, researchers developed new approaches to address issues such as binary code decompilation [9], de-obfuscation, anti-analysis, similarity checking [10] and detection [11].

Recently, in the scope of a collaboration with a national operational security center (SOC), researchers were able to recover the encryption key used by the *SaveTheQueen* ransomware without the need for a thorough and complex analysis of the malicious binary. This was largely achieved using a tool developed by the authors, hosted and executed in HSL to track and document all system calls generated by the audited binary.Despite the fact that this one is highly protected (use of the shellcode2exe and ConfuserEx obfuscators, injection into the winlogon process, self-modifying code using the Donut

[2]http://dns-bh.sagadc.org/
[3]https://sslbl.abuse.ch/blacklist/
[4]http://iplists.firehol.org/

[5]http://port2dist.lhs.inria.fr/

tool), it was possible to follow the handling of the encryption keys, particularly when importing them into the secure manager provided in Microsoft's Windows environment.

C. Secure hosting

We are also hosting tools and systems that operate on private data collected through experiments or by running services.

1) Voting system: A research group is developing Belenios [12] which aims at providing an easy to use voting system, guaranteeing state-of-the-art security, namely vote privacy and verifiability. It can be used in many types of elections and referendums, ranging from scientific councils to sport associations. Belenios guarantees vote privacy and full verifiability, even against a compromised voting server, as no one can learn the vote of a voter, since the votes are encrypted and the decryption key is split and shared among several parties. Moreover, it provides end-to-end verifiability. Every voter can check that his vote has been counted, and only eligible voters may vote. This feature relies on the fact that the ballot box is public (voters can check that their ballots have been received), and that the tally is publicly verifiable (anyone can recount the votes). Finally, ballots are signed by the voter credential (only eligible voters are able vote). Belenios is composed of a voting protocol, which has been formally proven, and a voting platform hosted in the HSL, in order to ensure the security of the voting server, and minimize the risks of compromission, while building user confidence in the platform. In 2019, the voting platform has registered a total of 6 340 of ballots with a total of 20 444 users, for a total of 172 elections, 22 of them having more than 200 voters.

2) Internet access observatory: In the Betternet project [6], researchers intend to build and deliver a scientific and technical collaborative observatory to measure and improve the Internet service access as perceived by users. Since users now are always connected and access different services through different networks and devices at different locations, the project propose new original user-centered measurement methods that rely on social sciences to better understand Internet usage, and the quality of services and networks. The whole project's infrastructure is hosted in the HSL, and the experiments are built around the advantages and functionalities offered by the platform. To measure end-to-end Quality of user Experience (QoE), researchers rely on crowdsourcing via the ApiSense [7] and Acqua [13] mobile applications deployed on end-users' smartphones. Relying on the HSL infrastructure, and the guarantees it offers regarding data confidentiality, is a key argument to convince users to share their data with the researchers, as well as the ethical committee who validates and authorizes the experimentations. In addition to measuring end-to-end QoE, they have to diagnose the sources of degraded experience and understand how users deal with this degradation. The HSL infrastructure acts as a ground truth reference during these experiments, in order to have reliable and repeatable measurements, on which researchers can apply controlled perturbations to validate the different detection algorithms implemented.

[6]https://project.inria.fr/betternet/

[7]https://apisense.io/

IV. Lessons learned

We successfully used the HSL facility for many years while researchers have conducted multiple projects and experiments in a safe and controlled environment. Below, we provide an overview of lessons learned from different perspectives.

A. Researcher experience

One of the most useful features for researcher while running their experiments is long-term data collection, i.e. keeping the historical data and collecting datasets over long time period. For example, when analyzing network traffic, in particular with machine learning methods, datasets preparation and annotation is one of the most important tasks. It is mandatory to classify the traffic, mostly the IP addresses, in different categories. When capturing data in realtime, or near realtime, it is very difficult to predict what information will be useful, and how it should be enriched or annotated. We tend to store raw data in PCAP format, and annotate the packets later on, by using public resources or databases such as blacklists of IPs known to be part of malicious activities. But due to the dynamicity of the Internet, if this processing is performed even only a few days after the capture, it can result in a biased and erroneous dataset. If we keep historical data, in particular the history of status changes for IPs in the blacklists, we can annotate each packet with the information relative to the time it took place, and not relative to the time the enrichment process is performed, resulting in a more accurate dataset. In addition, when evaluating detection and classification algorithms, we rely on the literature and news to create datasets of traffic annotated with known attacks. To create these datasets used for validation, we can either create them via simulation or in-lab experimentations, but these datasets end up being biased and not fully representative of the real world. By keeping historical data, we can create datasets from real network traffic, and thus have better data quality.

Another main objective of the infrastructure is to conduct experiments at large scale in a controlled environment. Most of them are sensitive and potentially disturbing, thus requiring the oversight of an ethical committee. Working in a controlled environment, where we can monitor all components of the infrastructure and running experiments, detect potentially harmful behavior and act in realtime is a key feature to obtain the validation of the committee for the experimentation.

Reactivity is a key factor in cybersecurity, in particular when a malware is discovered or a new attack is running in Internet. Not only is it important to ensure that our experimentations do not have harmful consequences, but also being able to deploy as quickly as possible probes or services to conduct such an experiment is a major concern. Relying on the HSL, and the fact that the security, monitoring and control of all components is ensured at the platform level, greatly reduces

the time required to become operational, as the majority of the required building blocks are already available.

B. Managing participants and resources

Data confidentiality is the objective of the infrastructure, allowing to work on sensitive data without compromising their security. This is a key argument when building partnership and cooperations with other organisations, universities or enterprises, as it helps to build trust and confidence. More precisely, as the HSL systematically relies on network isolation and access lists (ACLs) fine tuned at multiple levels (VPN, FW, LDAP, filesystem), we can control resources and data access with small granularity, and mutualize servers amongst several less sensitive projects from the same research group while ensuring confidentiality and most importantly integrity of data and processes. In addition, having a full control over the infrastructure, including user accounts and VPN profiles, ensures maximum reactivity when granting or revoking accesses. But this can also be a double-edged sword, as projects often require access to a trust zone for their interns or visitors, but very rarely notify when these accounts are not needed anymore, resulting in ghost accounts and VPN profiles that represent a security risk for the data of these projects. It is preferable to rely on expiration dates for accounts validity and maintain up-to-date a users database or listing to make sure accounts or permissions that are not required anymore are suspended, and VPN profiles revoked.

Applicative accounts are usually quite difficult to obtain, as policies usually involve a restriction to nominative accounts. The HSL's secure infrastructure and fined tuned ACLs allows us to offer this functionality, which can be very useful to deploy applications or probes at large scale under the same LDAP account but with different authentication keys for example. This allows us to control both individually and globally distributed resources.

We also recommend constant monitoring of hosts, systems and services, especially disk space. Our golden rule is to never trust users, but we usually need to give them more freedom and rights than we would like to, so that they can work freely in their designated areas. Relying on an efficient and constant monitoring and configuration is the key to strike a balance between permissibility and restrictions.

C. Takeaways and future work

We managed to operate and run for over 10 years the platform while relying on a reduced staff (less than one person full time). Using automated configuration and deployment solution, with double security for data (backup, raid with battery, redundancy when possible), allow us to operate and maintain the platform with minimal crew, and with few manual operations. This permits to have an efficient disaster recovery procedure allowing us to go back online quickly after a crash, and provide the services needed by the hosted projects with the required reactivity. So far, no critical data loss or compromission have been registered, despite the many projects the HSL hosted over the years.

Our platform is constantly updated to meet and plan one step ahead the requirements of researchers. In the near future, we plan several upgrades to cope with the technological needs of hosted projects, including internal connectivity upgrade to a 10GB network to allow better throughput between nodes in distributed environments or clusters. We would like also to add per node electrical monitoring and control, allowing us to monitor more precisely the electrical consumption and minimize it by remotely stopping unused nodes and starting them at-demand. In order to increase even more the security of the hosted data, we want to offer the possibility to research projects to rely on cryptography in an automated way, easing the deployment and usage of external backups. We also want to open the infrastructure to enterprises, either via different online services in SaaS mode, or via the open data initiative by deploying a dataset repository.

Acknowledgments

The authors wish to thank the team members of Belenios and Carbone for their valuable feedback on this paper. This work was supported in part by the CPER funding, and the ThreatPredict project funded by NATO.

References

[1] J. Mirkovic and T. Benzel, "Teaching cybersecurity with deterlab," *IEEE Security Privacy*, vol. 10, no. 1, pp. 73–76, 2012.

[2] P. Pfister, M. L. Wymore, D. Jacobson, and D. Qiao, "Design and implementation of a cyber physical testbed for security training," in *12th USENIX Workshop CSET*, Aug. 2019.

[3] A. Ashok, S. Sridhar, T. Becejac, T. Rice, M. Engels, S. Harpool, M. Rice, and T. Edgar, "A multi-level fidelity microgrid testbed model for cybersecurity experimentation," in *12th USENIX CSET*, Aug. 2019.

[4] "Caida project," https://www.caida.org/home, Accessed on 2020-04-13.

[5] J. Calvet, C. R. Davis, J. M. Fernandez, J.-Y. Marion, P.-L. St-Onge, W. Guizani, P.-M. Bureau, and S. Anil, "The case for in-the-lab botnet experimentation: creating and taking down a 3000-node botnet," in *Annual Computer Security Applications Conference*, Austin, Texas, United States, Dec. 2010.

[6] M. Coudriau, A. Lahmadi, and J. Francois, "Topological Analysis and Visualisation of Network Monitoring Data: Darknet case study," in *8th IEEE WIFS workshop*, Abu Dhabi, United Arab Emirates, Dec. 2016.

[7] S. Lagraa and J. Francois, "Knowledge Discovery of Port Scans from Darknet," in *IFIP/IEEE AnNet workshop*, Lisbonne, Portugal, May 2017.

[8] L. Evrard, J. François, and J.-N. Colin, "Attacker Behavior-Based Metric for Security Monitoring Applied to Darknet Analysis," in *The 16th IFIP/IEEE Symposium on Integrated Network and Service Management*, Washington DC, United States, Apr. 2019.

[9] G. Bonfante, J. Fernandez, J.-Y. Marion, B. Rouxel, F. Sabatier, and A. Thierry, "CoDisasm: Medium Scale Concatic Disassembly of Self-Modifying Binaries with Overlapping Instructions," in *22nd ACM Conference on Computer and Communications Security*, Denver, United States, Oct. 2015.

[10] G. Bonfante, J.-Y. Marion, F. Sabatier, and A. Thierry, "Code synchronization by morphological analysis," *7th International Conference on Malicious and Unwanted Software (Malware 2012)*, Oct. 2012.

[11] G. Bonfante, J.-Y. Marion, and F. Sabatier, "Gorille sniffs code similarities, the case study of Qwerty versus Regin," in *IEEE Malware Conference*, Fajardo, Puerto Rico, 2015.

[12] V. Cortier, P. Gaudry, and S. Glondu, "Belenios: a simple private and verifiable electronic voting system," in *Foundations of Security, Protocols, and Equational Reasoning*. Springer, 2019, pp. 214–238.

[13] O. Belmoukadam, T. Spetebroot, and C. Barakat, "ACQUA: A user friendly platform for lightweight network monitoring and QoE forecasting," in *3rd International Workshop on Quality of Experience Management*, Paris, France, Feb. 2019.