Towards an Infrastructure for Automated Distribution of Vulnerability Knowledge

Christian Kreibich, Jon Crowcroft
University of Cambridge Computer Laboratory
JJ Thomson Avenue, Cambridge CB3 0FD, United Kingdom
{firstname.lastname}@cl.cam.ac.uk

***Abstract*— The Internet is currently lacking an infrastructure that automates the distribution of new vulnerability knowledge to organisations and the (semi-)automated implementation of this knowledge within organisations. In this paper, we present an architecture that achieves this, the major challenges and ways to address them, and the implications of such an infrastructure for both security officers and attackers.**

***Index Terms*— security infrastructures, intrusion detection, attack description languages, vulnerability databases.**

I. Introduction

Over the past decades, security researchers have come up with a variety of devices designed to protect computer infrastructures. Examples include but are not limited to firewalls, proxy servers, vulnerability scanners, event and status loggers, intrusion detection systems (IDSs), and honeypots. Each of these Security Infrastructure Appliances (SIAs) focuses on a different aspect such as traffic filtering, intrusion detection, or vulnerability analysis. At present, there are a number of evident problems in the usage of SIAs:

- There is no standardised machine-processable attack description language, and no automated process to distribute new attack knowledge.
- All SIAs can produce a large amount of output that presents a burden instead of helping the analyst understand the activities on the network.
- Individual SIA output, or the lack thereof, is inconclusive since no device can guarantee by itself that a security breach has or has not occurred.
- SIA output correlation is inflexible. Automated log file correlation, if done at all, is currently a relatively inflexible, unidirectional process that does not allow the administrator to incorporate her knowledge fully.

In this paper, we propose an framework for the operation of computer security infrastructures that addresses these problems. Section II explains the current problems in detail. Section refsec:architecture then describes our proposed architecture, followed by a discussion of the benefits and dangers in Section IV. Finally, Section V summarises the paper.

II. Background

Computer security infrastructures are plagued by a number of problems that do not allow the deployed SIAs to be used to their full potential.

First, administrators need to pay attention to a number of vulnerability announcements and implement new advisories manually; this requires a great deal of knowledge, manual labour, and discipline to be done correctly. Administrators have to follow the published updates, determine which updates are relevant to their organisation, download required software patches, and update the affected systems properly. The Internet currently lacks a standardised, machine-processable attack description language that precisely describes a vulnerability, the pre- and postconditions of a successful exploit, and the sequences of steps an attacker has to perform to succeed. Similarly missing is a mechanism that allows easy integration of of such attack knowledge into existing security infrastructures. Signature repositories of misuse-based IDSs are not helpful in this regard: IDSs typically need precise tuning to be useful; simply downloading and installing new signatures will often only produce large numbers of irrelevant, erroneous alerts.

This is the gist of the second problem: All types of SIAs generate output that has to be inspected by the analyst in order to gain maximum benefit from employing those appliances. This output can quickly become so voluminous that it effectively becomes noise, and important information is drowned in the flood. IDSs are notoriously bad in that regard: The number of alerts that IDSs signal on busy sites when they are not tuned properly can go into the tens of thousands per day. Output of this magnitude is practically useless without appropriate post-processing. Likely, most of the alerts are *false positives* that occur when the IDS erroneously declares a harmless event to be security-relevant. This phenomenon is currently the most prominent problem of IDSs and cannot be entirely eliminated with even the best of tuning.

Third, an alert signalled by an individual SIA does not necessarily mean that a network is under attack, and the absence of an alert does not mean that it is not. IDSs are unlikely to detect all malicious activity—particularly network-based IDSs are prone to being evaded [1][2][3], producing *false negatives*. Relying solely on the performance of individual SIAs cannot be a sound basis of an incident response scheme. Unfortunately, this is common practise right now. Effective event correlation is crucial for meaningful alert reports, as it can increase the reliability of the resulting alert.

Fourth, while correlation can significantly improve the quality of alerts, this correlation is not yet mature practise and provides only a fixed set of reduction mechanisms (e.g., filtering by source or destination IP addresses). In particular, the administrator is not given mechanisms to *prevent* false positives in the future and prioritise alerts—the state of the art is only an a-posteriori *filtering* mechanism.

III. Proposed Architecture

We propose a framework for the operation of security infrastructures that addresses each of the problems explained in the previous section. Our framework uses the following components:

A **machine-processable attack description language** that allows the precise definition of vulnerabilities, taking into account sequences of events as well as pre- and postconditions of steps required for an attack that leads to successful exploitation of the vulnerability. These definitions must not be formulated in terms of a single SIA's output (particularly, not only IDS signatures) but at a higher level, *abstracting* from the characteristics of actual SIAs. Note that we do not propose a language that describes how to *perform* an exploit; attack descriptions only describe the necessary steps and the observable characteristics when these steps are performed.

A number of schemes to define attacks have been proposed in the literature that could be leveraged [4][5][6][7]. What is missing in these approaches is a commonly-accepted, easily-processable format and an accepted taxonomy to refer to. Recent work on attack ontologies may be usable in this context [8]. We propose an attributed version of attack trees, defined in a semantically strict language and distributed in a standardised form such as XML Schemas, to allow flexible processing. Attributing the stages of a complex attack based on an ontology that specifies the semantics of possible subgoals (e.g., in terms of having access to a port, obtaining a user account, or executing programs) enables automated transformation of these stages to configuration items of SIAs (see Figure 1).

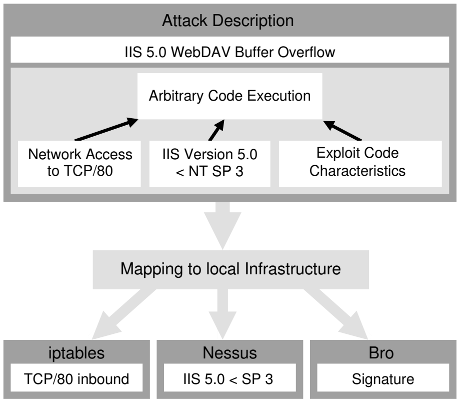

Fig. 1. Mapping an Attack Description to Configuration Items.The attack description contains information about an IIS buffer overflow which allows attackers to execute arbitrary code on the victim machine. A successful attack requires access to TCP port 80, a vulnerable version of IIS and the transmission of the exploit code. In the mapping process, the port access is mapped to an iptables firewall configuration item, the server version becomes a Nessus vulnerability scanner item, and the exploit code characteristics are described by a Bro IDS signature.

An **online repository of attack descriptions** that allows individuals to upload attack descriptions so that other organisations can easily access them. Uploaded descriptions are signed to indicate their originator. The repository provides mechanisms for browsing the attack descriptions (e.g., to allow searching by OS type or vulnerable software versions) and for rating the quality of existing descriptions. More importantly, a subscription mechanism allows organisations to request relevant attack definitions to be pushed to their security infrastructure automatically.

The transport mechanism for the attack descriptions must guarantee reliability, integrity and confidentiality. The BEEP [9] tunnel and transaction layer security profiles, also used in the Intrusion Detection Exchange Protocol [10], provide these features.

Two existing schemes are conceptually related to automated distribution systems for vulnerability knowledge. The distribution channels of anti-virus signatures are user-initiated and more simplistic than attack descriptions. The software package management systems of the major open-source OS distributions allow users to check for updated versions of packages and then initiate upgrades. There are currently no robust signature distribution schemes for IDSs.

Full integration of the human element. For the foreseeable future, humans will be needed to properly analyse the activities reported by the SIAs [11]. Determining whether activity is malicious or not is essentially a classification process, therefore tuning SIA operation should be considered a *learning process*. Administrators often possess solid experience in detecting intrusions. The administrator must be fully integrated in a feedback loop to make informed decisions about the quality of reported alerts and to then *teach* the SIAs how to behave in the future. The administrator needs to be able to define precise abstractions (detection *patterns*) to reduce false positives and to rank reported alerts as desired. This is essentially a pattern matching application on the features extractable from correlated SIA output data. SIA output correlation is still immature and mostly focuses exclusively on IDS output. Previous work in the field [12][13][14][15][16] uses a variety of approaches to correlate the data, but the approaches are generally single-directional and do not allow the administrator to incorporate her knowledge in the decision-making at the sensors to reduce the amount of reported low-level alerts in the future.

The management environment must also allow semi-automated[1] incorporation of attack descriptions obtained from an online repository into the local infrastructure. The main difficulties here are the threat of triggering large amounts of false positives, and the danger of hindering legitimate activity. Several concepts could help mitigate these risks. First of all, the administrator must be able to express which organisations are trusted to provide reliable attack descriptions. Furthermore, the application of new descriptions to the local infrastructure has to be guided by a mission–impact approach [17]: A severity metric could be used to limit the effects of a new attack descriptions on the local infrastructure—drastic measures like blocking traffic only get applied when the administrator sufficiently trusts the originator and when the severity of the new attack is higher than the availability requirements of the affected service, as specified by the administrator.

An overview of the architecture that puts these components into context is shown in Figure 2.

IV. Discussion

The proposed framework offers significant benefit: Automating and integrating the acquisition, application, and dis-

[1] We use the term "semi-automated" to mean a principally automated, but optionally human-guided activity.

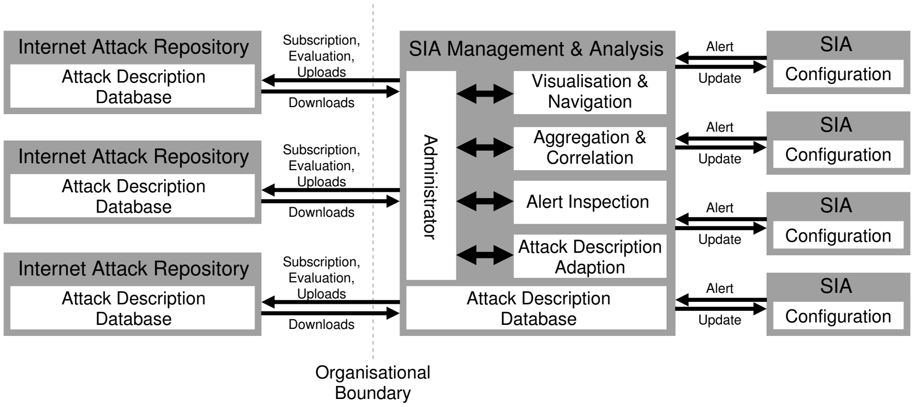
Internet Attack Repository
Attack Description Database
Internet Attack Repository
Attack Description Database
Internet Attack Repository
Attack Description Database
Subscription, Evaluation, Uploads
Downloads
Subscription, Evaluation, Uploads
Downloads
Subscription, Evaluation, Uploads
Downloads
Organisational Boundary
SIA Management & Analysis
Administrator
Visualisation & Navigation
Aggregation & Correlation
Alert Inspection
Attack Description Adaption
Attack Description Database
Alert
Update
Alert
Update
Alert
Update
Alert
Update
SIA
Configuration
SIA
Configuration
SIA
Configuration
SIA
Configuration

Fig. 2. Architecture of the framework.

tribution of vulnerability knowledge would significantly ease the application of well-defined processes and policies in an organisation's computer infrastructure. The current state of the art imposes limitations on both the speed and the efficiency with which new vulnerability knowledge can be applied.

However, the idea also raises the question whether such a framework could be abused by attackers. Providing substantial amounts of security-relevant knowledge in central repositories will undoubtedly attract attackers. The primary points of concern are as follows:

- Leveraging provided knowledge: Attackers could try to use the provided knowledge to write new exploit code. This can be prevented by focusing on the vulnerability and not the means to exploit it, as the vulnerability descriptions only describe the steps involved in a successful exploit, but not the ways to perform this. We believe that the vast majority of attackers do not have the skills to do this and rely on code provided elsewhere for this purpose.
- Uploading malicious information: Attackers could attempt to upload fake vulnerability descriptions that result in behaviour in the SIAs that attackers can exploit to their advantage, such as large numbers of false positives. We believe that the feedback and rating mechanisms could prevent attacks of this kind.

Another point to consider is the willingness of organisations to contribute new vulnerability knowledge. Traditionally, organisations have been hesitant to publish information regarding security breaches that occurred in their own infrastructure. This issue is not relevant here, as the uploaded descriptions are instantly useful to other organisations. Therefore, we believe that organisations will use this framework as a way to build up or strengthen their reputation in security-awareness.

The architecture relies on a standardised format to describe vulnerabilities and the conditions that need to be fulfilled to exploit them. We believe that a standardisation process for such a language would enable significant advances in the network security domain, as pointed out by our framework.

V. Summary

We have proposed a framework for automated distribution and application of vulnerability knowledge. The framework relies on three core functionalities. The first is a machine-processable attack description language that describes vulnerabilities and the stages involved in exploiting them. The components of attacks are categorised by referring to a vulnerability ontology, to allow automated transformation to configuration items of security infrastructure appliances. The second component is online repositories of attack descriptions that allow organisations to publish new descriptions, to subscribe to a selective distribution system, and to rate existing descriptions. Finally, the third component is better integration of the human element into the security infrastructure. The administrator must be able to supervise the application of new vulnerability knowledge to the local infrastructure, and define precisely the abstractions that need to be made from the reported activities to yield useful alerts.

The framework integrates and automates several of the mission-critical tasks in running a computer security infrastructure. Care needs to be taken to prevent individuals from using the infrastructure for malicious purposes; however, we believe that these risks can be mitigated.

References

[1] T. H. Ptacek and T. N. Newsham, "Insertion, Evasion and Denial of Service: Eluding Network Intrusion Detection," Secure Networks, Inc., Tech. Rep., 1998.

[2] M. Handley, C. Kreibich, and V. Paxson, "Network Intrusion Detection: Evasion, Traffic Normalization, end End-to-End Protocol Semantics," in *Proceedings of the 9th USENIX Security Symposium*, 2000.

[3] U. Shankar and V. Paxson, "Active Mapping: Resisting NIDS Evasion without Altering Traffic," in *Proceedings of the 2003 IEEE Symposium on Security and Privacy*. IEEE, 2003, pp. 41–59.

[4] B. Schneier, *Secrets and Lies*. New York: John Wiley and Sons, 2000.

[5] T. Tidwell, R. Larson, K. Fitch, and J. Hale, "Modeling Internet Attacks," in *Proceedings of the 2001 IEEE Workshop on Information Assurance and Security*. United States Military Academy, West Point, NY: IEEE, June 2001, pp. 54–59.

[6] A. P. Moore, R. J. Ellison, and R. C. Linger, "Attack modeling for information security and survivability," Carnegie Mellon University, Tech. Rep. CMU/SEI-2001-TN-001, 2001.

[7] S. Eckmann, G. Vigna, and R. Kemmerer, "Statl: An attack language for state-based intrusion detection," Dept. of Computer Science, University of California, Santa Barbara, Tech. Rep., 2000.

[8] J. U. J. Pinkston, "Modeling Computer Attacks: A Target-Centric Ontology for Intrusion Detection," 2002.

[9] M. Rose, *The Blocks Extensible Exchange Protocol Core, RFC 3080*, 2001.

[10] Internet Engineering Task Force, Intrusion Detection Working Group, "Intrusion Detection Exchange Format," http://www.ietf.org/html.charters/idwg-charter.html, 2000.

[11] J. Allen, A. Christie, W. Fithen, J. McHugh, J. Pickel, and E. Stoner, “State of the Practice of Intrusion Detection Technologies,” Carnegie Mellon University, Tech. Rep. CMU/SEI-99-TR-028, Jan. 2000.

[12] G. Vigna, R. Kemmerer, and P. Blix, “Designing a Web of Highly-Configurable Intrusion Detection Sensors,” in *Proceedings of the 4th Intern. Symposium on Recent Advances in Intrusion Detection (RAID)*. Springer Lecture Notes in Computer Science 2212, 2001, pp. 69–84.

[13] A. Valdes and K. Skinner, “Probabilistic Alert Correlation,” in *Proceedings of the 4th Intern. Symposium on Recent Advances in Intrusion Detection (RAID)*. Springer Lecture Notes in Computer Science 2212, 2001, pp. 54–68.

[14] W. Lee, S. J. Stolfo, and K. W. Mok, “A data mining framework for building intrusion detection models,” in *IEEE Symposium on Security and Privacy*, 1999, pp. 120–132.

[15] O. Dain and R. Cunningham, “Fusing a heterogeneous Alert Stream into Scenarios,” in *Proceedings of the 2001 ACM Workshop on Data Mining for Security Applications*, 2001, pp. 1–13.

[16] B. Morin, L. Me, H. Debar, and M. Ducasse, “M2D2: A Formal Data Model for IDS Alert Correlation,” in *Proceedings of the 5th Intern. Symposium on Recent Advances in Intrusion Detection (RAID)*. Springer Lecture Notes in Computer Science 2516, Oct. 2002, pp. 115–137.

[17] P. A. Porras, M. W. Fong, and A. Valdes, “A Mission-Impact–Based Approach to INFOSEC Alarm Correlation,” in *Proceedings of the 5th Intern. Symposium on Recent Advances in Intrusion Detection (RAID)*. Springer Lecture Notes in Computer Science 2516, Oct. 2002, pp. 95–114.