COMS30038 — SECURITY BEHAVIOURS

THREAT MODELLING: SECURITY OPERATIONS

*Matthew Edwards**

---

1 SECURITY OPERATIONS

It would be a far simpler world for computer systems defenders if computer security was 'just' a case of properly installing the correct protective measures[1]. An organisation in this fantasy world could hire a fully-credentialed security specialist to come install the proper protections, hand them a hefty cheque, and then never have to think about computer security again. Unfortunately, it turns out that preventing people from breaching your system's security is next to impossible. Given that you also want people to use your system, and this often involves adding new devices, connections, software and accounts, there are always new routes around protective measures.

As we cannot be sure of any passive measure keeping attackers 'out', an important component of security work is detecting intrusions into your system – ideally, as they are happening, but also after the fact, so you can trace the attackers' method and patch the hole they found. The necessity of system surveillance for threat detection was first articulated 40 years ago [1], and since then intrusion detection has come into its own as a field of cybersecurity research, with its own fundamental models [2], and several million publications proposing alternative approaches, new techniques, simulations, empirical evaluations and comparisons. This lecture doesn't aim to summarise this literature, but to provide you with a model for the process of intrusion detection, and how we acquire and update knowledge about threats.

A key lens for this topic is to view cybersecurity as an *autonomic computing* application area [3]. Autonomic computing focuses on the creation of computer systems that can adapt to changing environments, with the aim of making them self-managing. A core concept of any autonomic computing system is a 'control loop' that describes how the system monitors and responds to its environment – and something you definitely want in cybersecurity is to detect and react to attacks. A straightforward control loop model that can be

*matthew.john.edwards@bristol.ac.uk

[1] Admittedly, 'properly' and 'correct' are covering up a lot of work in that sentence.

applied to cybersecurity is the MAPE-K model [4]. The name stands for the five core components of the control loop: *Monitor*, *Analyse*, *Plan*, *Execute* and *Knowledge*. This is a general-purpose autonomic systems model, as might be used in something like designing an autonomous robot, but here we'll apply it to organisational cybersecurity in particular.

1.1 *Monitor*

Detecting an attack on your system is only possible if you are monitoring that system. This means that the security team needs to have *sensors* within the system that are capable of observing its behaviour, ideally at a highly-granular level of detail. This can in itself be a fraught issue, as such sensors extend a powerful surveillance architecture over the system, which might itself be perceived as a security threat. Consider the perverse incentives in the example of a security administrator who is able to monitor every keystroke of his superior, or the payroll team, or even the CEO[2]– the temptation exists to pry into matters above your official clearance level, or even to gather blackmail material. Even more worryingly, the existence of such sensors could itself be vector for external attackers to obtain sensitive information[3].

Leaving these big-picture questions aside, we can focus on the question of what exactly should be monitored. While this will necessarily depend to some degree on the specifics of the organisation being monitored, there are three key targets for automated monitoring:

Network monitoring is critical, in part because networking itself is so critical to modern infrastructure. Most attacks will arrive via a network link. One of the best kinds of network information is the *full packet capture* – a complete transcript of network activity over a monitored link. However, storage limitations mean that network sensors cannot retain large windows of packet capture data, and instead must filter a small selection of monitored traffic to report when a potential security event is detected. Sometimes network attacks are best detected at an aggregate traffic level, and, for large networks, such network 'flow' data may be the only practical method of performing scalable monitoring [7].

Host monitoring involves operating system and kernel logs, which record the various system calls being executed by processes. System call monitoring is critical to the detection of active malware, and a key part of most antivirus systems, as well as providing much insight into the activities of malicious actors within a compromised host. This logging can often be difficult to erase from userspace, making it a valuable post-breach

[2]The issues I mention here are core to the area of *role-based access controls*, which we don't cover in detail in this course. If you're interested, a good starting point might be the canonical publication by Sandhu et al. [5].

[3]There are an interesting array of other possibilities for evading or exploiting intrusion detection systems, discussed in some detail by Lyon [6].

source of evidence. The `syslog` logging infrastructure is heavily-used for consolidating host-based monitoring for security purposes.

Application monitoring provides a higher-level view of events which can be more easily interpreted. Application logs were initially created to aid in debugging and system management, and so they produce more textual, helpful messages. Examples of application logs commonly involved in monitoring are web server logs and remote login logs[4]. Both identify the various requests fielded by the server, along with the response sent.

A fourth strand to monitoring which is sometimes neglected is human reporting. The everyday users of a system become much more intimately familiar with it, and can be the first to identify bugs or strange new behaviour that may indicate a security breach. Making it easy for users to report potential problems can provide security operations staff with invaluable on-the-ground intelligence. From the perspective of human behaviour, the critical questions in monitoring tend to be related to getting users to engage with these reporting systems.

1.2 *Analyse*

Given the number of low-level events generated by monitoring even quite a small system, manual review very quickly becomes impractical for all but the most critical investigations. As such, security operation centres rely heavily on automated approaches to analysis of security events, with intelligent tool support for this being correspondingly a research focus. There are two major approaches to this: misuse detection and anomaly detection.

Misuse detection relies on automatic detection of behaviour known to be malicious in nature. For example, many types of malware have specific execution patterns—a particular series of system calls—which can be identified in host-based logging. Similarly, many bots and worms present particular network traffic profiles, or dial out to known 'bad' IP addresses. By capturing and sharing these indicators—sometimes referred to as *indicators of compromise* or IoCs—organisations can improve their ability to detect and prevent cyber-attacks. The drawback of this approach is it requires these existing fingerprints in order to detect malicious behaviour, and so is usually unable to identify novel attacks as they are happening.

Anomaly detection takes a different approach. Rather than detecting known malicious behaviour, anomaly detection takes the position that strange new behaviour of a system is likely to be malicious—or at least worth investigating—and so focuses on identification of deviations from 'normal' system behaviour. This is established by creating a baseline for the expected state of the system, usually by observing 'clean' logs and generalising from those. The benefit of this approach is that it can detect a great deal of malicious behaviour without having previously encountered it – covering the hole that

[4] e.g., for `ssh` connection attempts.

misuse detection leaves open. The drawback is that the threshold for anomalous activity can be hard to set. If the threshold is set too high, attacks will slip through. If it is set too low, then even minor deviations from what was seen in the 'normal' logs will be flagged as security incidents, swamping the human analysts with low-value alerts.

This last problem is already a concern in this area. Security operations staff are often flooded with spurious alerts, which take up a lot of their time and make it more likely that they will miss or be unable to respond to genuine threats in a timely manner. Part of the solution here is purely technological – better models for detecting real threats. But another part is the consideration of the human-computer interaction going on in these centres: the display mechanisms, the workflow integration, and how automation can best integrate with security management tasks.

1.3 *Plan*

The presentation of alerts to human staff moves the cybersecurity control loop into the *Plan* stage, where the system typically relies most heavily on its human components to direct responses to the environment. The actual responses are covered within the *Execute* phase, so here we'll discuss some of the longer-term planning that needs to be accomplished within the loop.

Primarily, security management needs to understand the risks they are dealing with in their organisation . There are a number of formal models for thinking about security risk, but some typical approaches involve identifying the organisation's critical assets[5], and then a combination of the risks inherent to their sort of business, the countermeasures already in place, and the particular threat profile the organisation has faced in the past and might still be facing [8]. As all of these factors will shift over time, maintaining an appropriate risk assessment and mitigation strategy is also an ongoing process, requiring reassessment based on both what is seen from monitoring and what can be learned from other sources about relevant risks. Of course, there are other entities that might be interested in modelling your risk of suffering an attack – like your cyber-insurance providers.

There are also a number of more traditional management tasks that need to be accomplished in maintaining a system's security. An important area for consideration here is performance appraisal. Taking the outside view, imagine you are the management for a company which has outsourced its security operations to a third party, as is quite common. Now imagine that you experience no or very few security incidents over a year. Is this because the security operations centre is performing well at dealing with them, or is your company just not suffering attacks? The next fiscal year is going to

[5]Not always as straightforward as you might expect. Often with small companies there is a lot of experimental restructuring and use of external services, and it can remain non-obvious to the company exactly how much they rely upon something until it suddenly stops working one day.

be tight, can we cut costs by dropping this service? Of course your service provider is going to *say* they're essential to your business, but how can you trust them? Conversely, imagine that your company suffered a great many security incidents this past year. Can you conclude that your SOC is rubbish and you need to find a better one, or is it actually that your organisation being heavily targeted, and you're seeing only the few rare lapses in some impressive defences? To help tackle these problems of benchmarking security, some industry-wide standards have been adopted for performance measurement, which focus on assessing the state of preparedness and formal processes for dealing with various threats [9]. Building these processes is an important part of the planning stage of the cybersecurity control loop, and critical to having them in place so they can be swiftly enacted in the case of an intrusion.

1.4 *Execute*

What the appropriate response for an attack is depends to some degree on whether it is an attack that has been detected while in-progress—in which case action can be taken to *prevent* the attack from completing—or after the fact, when the response needs to focus on appropriate *recovery* of the system back to a secure state of operation.

Prevention responses typically involve resolving incidents by creating targeted countermeasures for the detected malicious behaviour. This might mean blacklisting an IP address, killing a process, or temporarily halting the operations of an application that has been subverted and is leaking data to attackers. As many attacks are automated, timely intervention in attacks is often beyond human capabilities if they are not sustained. As such, many intrusion detection systems have been augmented with prevention capabilities, creating detection tools capable of automatically blocking what they perceive as high-probability attacks at the same time as they produce an alert about it. These systems are known as intrusion prevention systems.

Recovery responses to attacks involve rebuilding your security—and perhaps the organisation's ability to work at all—after a breach has occurred. Depending on the focus of the attack, and the measures put in place beforehand, this might involve different activities. A ransomware attack might mean restoring the system state from a backup, so long as you have been backing-up your data[6]. A data breach might mean notifying your customers, and potentially paying out fines to a regulator[7]. In most cases, however, an investigation is required in order to establish

1. How the attackers got into your system, so you can increase protective measures. This can be fairly important, as news of your

[6] And so long as the ransomware hasn't also reached your backups.
[7] Possibly covered by your cyber-insurance, if you have it.

vulnerability can cause other attackers to probe your system. A script kiddy managing to deface your website might not themselves cause much damage, but the persistent threat that follows them through the same exploit could well be much more of a problem.

2. Whether the attackers left any backdoors that would allow them to re-enter your system after restoration. It is no good fixing the hole they used to get in the first time if you don't notice the key they left themselves to come back whenever they want. Typically a full audit is required.

1.5 *Knowledge*

Finally, the cybersecurity control loop needs to integrate knowledge from outside the system being protected. This is how the misuse detection systems can receive new indicators of compromise to watch out for, and how the security team can adjust their understanding of the risks their system is exposed to, in order to plan the appropriate responses *before* the attack lands. There are a number of common sources for this information.

Cyber Threat Intelligence organisations run honeypots to identify threats and capture the indicators of compromise (IoCs) related to attacks. These can then be fed into misuse detection systems to inoculate them against attacks before they strike.

Computer Emergency Response Teams also share information on threats, more typically as higher-level advisories, but they also share common best practices for a sector or organisation.

Common Vulnerabilities and Exposures are best known as CVEs, and provide a reference dictionary for known software vulnerabilities, referencing specific versions of systems, and even carrying a severity rating for each CVE under a common vulnerability scoring system ranging from 1-10. You can access the public CVE database at `https://cve.mitre.org/`.

Common Attack Pattern Enumeration and Classification also known as CAPEC, is a large, up-to-date dictionary of high-level attack patterns which adversaries are known make use of, available at `https://capec.mitre.org/`.

Adversarial Tactics, Techniques & Common Knowledge also known as ATT&CK, is a reference framework that describes in detail some of the behaviour you might expect of attackers as they move through your system, with examples of how known threat actors performed these attacks. You should recognise the matrix organisation of the framework – it's a killchain model. The framework is available at `https://attack.mitre.org/`.

2 OPTIONAL EXERCISES

These are suggestions for students who are particularly interested in the topic of investigating attacks as a system operator or security professional. They are not assessed in any way. You don't need to complete the below to do well on the course.

2.1 *Further Reading*

For those of you who enjoyed this week's reading [10] and would like more in the vein of narrative-driven, somewhat playful cat-and-mouse between a systems administrator and their pernicious foe, I can think of no better reading than the book *The Cuckoo's Egg* by Cliff Stoll [11]. A systems administrator for a laboratory discovers a 75-cent accounting error between two programs, and his quest to pin down why eventually leads him into a real-life transnational espionage sting involving the FBI, CIA, NSA and KGB. Despite being originally published in the late 1980s[8], the book outlines a number of enduring issues in cybersecurity, and is a very enjoyable read at the same time.

2.2 *Practice*

Computer security is a constantly-evolving area that touches on a wide array of technical topics. Even professionals working in the area need periodic training to stay up-to-date. Unfortunately, this is well known, and there is a predatory ecosystem of 'security training' consultancies who will charge you for some time with Powerpoint and give you a practically-worthless credential to append to your CV. The best remedy I can suggest for this is to be very *specific* about your training needs. Identify a topic you need to know more about, do some preliminary reading and only *then* look for a course or seminar that will give you the hands-on guidance you need. You are far less likely to be fleeced if you know what you want going in.

REFERENCES

References

[1] J. P. Anderson, "Computer security threat monitoring and surveillance," James P. Anderson Company, Tech. Rep., 1980.

[2] D. E. Denning, "An intrusion-detection model," *IEEE Transactions on Software Engineering*, no. 2, pp. 222–232, 1987.

[3] H. Debar, "Security operations knowledge area," in *Cyber Security Body of Knowledge*, 2019.

[8] I refer you to the updated 2005 edition with a good afterword.

[4] IBM, "An architectural blueprint for autonomic computing," *IBM White Paper*, vol. 31, pp. 1–6, 2006.

[5] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," *Computer*, vol. 29, no. 2, pp. 38–47, 1996.

[6] G. Lyon, "Detecting and subverting firewalls and intrusion detection systems," in *Nmap Network Scanning*, 2011. [Online]. Available: https://nmap.org/book/subvert-ids.html

[7] D. Rossi and S. Valenti, "Fine-grained traffic classification with NetFlow data," in *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference*, 2010, pp. 479–483.

[8] D. W. Straub and R. J. Welke, "Coping with systems risk: Security planning models for management decision making," *MIS Quarterly*, pp. 441–469, 1998.

[9] ETSI Industrial Specification Group, "Key performance security indicators (KPSI) to evaluate the maturity of security event detection," European Telecommunications Standards Institute, Tech. Rep., 2018.

[10] B. Cheswick, "An evening with Berferd: In which a cracker is lured, endured, and studied," in *Proceedings of the Winter USENIX Conference, San Francisco*, 1992, pp. 20–24.

[11] C. Stoll, *The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage*. Simon and Schuster, 2005.