Technical Exploit Management: You’re Doing It Wrong

The number and type of technical vulnerabilities that are being discovered or created is staggering. Whether attackers use unquoted paths for binary attacks, responder for NTLM attacks, SQL injection, .lnk attacks, print daemon attacks, or base64 encoded URL DNS exfiltration the technical exploit field of options is becoming a never ending plateau of poison. It is critical for organization to have a technical exploit management program to identify, triage, and respond to technical exploits that are known and even the ones that are unknown. In fact, a great number of organizations already have a technical exploit management program; however, these organizations do not realize they have chopped it up and embedded it in other initiatives like patching and network security. To ensure a unified understanding of what constitutes a technical exploit management program, for the purposes of this article, a definition will be provided along with the difference between known and unknown technical exploits. Once there is agreement about those terms, the attention will turn to how to discover technical exploits of concern. Once identified, technical exploits need to be responded to and the response does need to tailored to the type of technical exploit (known or unknown). To conclude this article a brief discussion of one way to practically implement a technical exploit management program.

Most organizations have a patching program and security devices that detect anomalous behavior and activities, receives OSINT alerts and intelligence, and even have solutions that will determine if any files were modified without permission but there is no unification of these efforts under one initiative that separates them from the rest of the information security or cybersecurity macro-programs. An “Exploit Management Program” is an initiative that uses open and closed source intelligence, network and host solutions, and vendor support to identify, respond, and reduce the risk to known and unknown technical exploits that are likely to have an impact on the corporate risk exposure metrics. Simply put, it is a program that identifies technical exploits that have been made public or are being used and then puts forth effort to reduce the risk of those exploits being successfully used. A known exploit is any exploit that has been publicly disclosed or for which a patch has been released. Unknown exploits are those technical exploitation methods and tactics that are not publicly released. The methods for identifying these two kinds of exploits is different but identifying known exploits is far more important as they are used in the majority of attacks as demonstrated by any breach report, exploit report, or coagulation of after-action-reports anyone can find and read.

Identifying known exploits is more formulaic and repetitive than entertaining for most professionals but it is an essential first step in the implementation of a technical exploit management program. The best ways to identify known technical exploits that are relevant to the enterprise come from three main sources. The first and easiest source is vendor patches or vulnerability scanners. There are few organizations that are not familiar with or whom do not use patches so it makes sense that they are already formally part of the enterprise security practices. When patches are released there will be

security patches with various risk ratings and organizations can further evaluate the risks based on the applicability to their environments. The second is professional alerts from security organizations and departments. While there are a great number of private organizations that provide threat data to the world (sometimes at considerable cost) there are government organizations that also provide security alerts. Some of the government run alerting entities include CISA from the United States (us-cert.cisa.gov), NCSC from the UK (ncsc.uk.gov), JPCERT from Japan (jpcert.or.jp), Canadian Centre for Cyber Security (cyber.gc.ca), and ACSC from Australia (asd.gov.au). There are many other countries (and some private entities) that have sources of alerts that are also invaluable so this list is just a starting place. The third place to to look for known technical exploits is in exploit publications databases such as metaploit and exploitdb. These databases do require more effort as locating exploits related to the organization will need to be located using scripts or human manual processes followed by evaluation to determine if the results are accurate and actually relevant. Identifying known exploits is far more important and much simpler than identifying unknown exploits and should always be fully implemented prior to moving toward identifying unknown technical exploits.

The identification of unknown exploits requires a fairly high level of program maturity and technical prowess. When a company reaches this level of maturity there are three ways in which they can pursue the identification of unknown technical exploits. These types of exploits also have three main avenues for discovery. The fist, and most common, way to identify unknown technical exploits is by being the victim of an attack. In this scenario an attacker uses a technical exploit which is not publicly known and the victim does a root cause analysis with detailed log data that supports the investigation to determine the specific bites sent to a specific piece of code. The organization then replicates the scenario in a lab setting to verify the unknown technical exploit. At this point the unknown technical exploit becomes known to the organization but unless that organization publicly discloses it, or reveals it to the vendor which created the exploitable software, the exploit remains an unknown technical exploit to rest of the community or world. The second way in which unknown technical exploits are identified is through the use of an internal exploit development team. This team can be part of an offensive red-team, a threat hunting team, a software quality assurance team, or any other team who is responsible for developing ways to compromise or test the technical security of a piece of software. The third and final way for organizations to identify unknown technical exploits is through reconnaissance activities. As some of these activities may not be lawful in certain jurisdictions it is prudent to consult the legal team before implementing any reconnaissance initiatives. In reconnaissance, the organization builds a presence on the “dark web” or infiltrates potential adversaries’ organizations. In the dark web scenario the organization’s presence is masked with a variety of counter intelligence, anti-identification, and anti-tracking techniques. Once a presence has been established with the supporting security infrastructure the organization builds a reputation with criminal or perceived-to-be criminal entities to gain trust. One way to do this is to (falsely) publicly disclose a breach that had no real impact other than a defacement of a minor subdomain then provide proof through the dark web persona that the defacement was performed by the dark web persona. As trust and reputation is built the organization may be approached with or invited to public or private sales where exploits are sold. These exploits are rather expensive and often have a money-back

guarantee. These exploits are unknown exploits. Once an organization does identify known and unknown exploits they need to triage.

Technical exploit response actions have two main choices, patch or mitigate. Some might consider accepting the risk as a reasonable third choice but this article disagrees with that stance as discussed at the end of this section. When a patch is available, has gone through the organizational patch process, and then applied to the affected systems there is no need to perform additional response activities. When a patch is available and has not, or can not, be applied the technical exploit must be mitigated. In technical exploit mitigation compensating controls need to be evaluated and applied. Often times, for known technical exploits, there are work-around processes that constitute compensating controls. When work-around solutions are not available the organization should develop a process to categorize, prioritize, and initialize a response. For the categorization portion, there are a number of category styles and types that can be used. This article uses infiltration, lateral movement, privilege escalation, and exfiltration. Each unpatchable exploit should be placed in one of these four categories and the categories should be assigned a metric that indicates the order of criticality. The order may be different for every organization; however, if an attacker cannot exfiltrate data the purpose for the attack is severely limited, likewise, if an attacker cannot gain privileges there should be rather limited attack possibilities. Once the technical exploits have been categorized they should be prioritized. Exploit prioritization most commonly takes the public CVSS score (if available) or creates one based on best effort then adjusts that score based on applicability to the enterprises current security mitigation efforts and environment. Once the categorization and prioritization have been completed, the scores for each are multiplied and the technical exploits are mitigated in order from highest-score to lowest. The simplest and most common mitigation technique is to increase logging and security alerting around that exploit using signatures or behavioral detection. This can be quite effective but if this is the only solution used it can create alert fatigue. Another approach is to block the ability to attack or reduce the surface area of the attack. This can be accomplished using network devices that limit whom can send particular traffic or by host software that limits whom can interact with particular software such as like preventing users from creating .bat files on a windows OS (https://community.spiceworks.com/78725-prevent-bat-files-for-users). It is nice to be able to conceptually identify, prioritized, and mitigated but putting these steps into a unified program can seem complex.

The majority of the technical exploit management program should exist in most organizations but may not have the inter-departmental or program communication to solidify a mature effort. Vulnerability identification is best suited for roles such as incident responder or threat intel specialist for known vulnerabilities with unknown vulnerability identification being best suited for technical threat hunters, senior security programmers, and senior threat intelligence operatives. Once the vulnerabilities are identified they should be placed in a central (highly restricted and protected) database where the discoverer (or a group) determines if the exploit would be most likely used for infiltration efforts much like a specific PHP injection (https://www.acunetix.com/websitesecurity/php-security-2), privilege escalation like Dirty Cow attacks (https://en.wikipedia.org/wiki/Dirty_Cow), lateral movement such as Kerberoasting (https://adsecurity.org/?p=3458), or exfiltration efforts as is the case with DNS

exfiltration (https://insinuator.net/2020/03/dns-exfiltration-case-study). Once the technical exploits are categorized (or using the same data entry worksheet) they also need to have a prioritization metric added. As the database is filled with exploits that have not been patched the most critical ones will have the highest score. At this point an organization should determine at which score level an exploit can be mitigated using alerts and which need compensating controls that limit the attack exposure. For alerts, signatures or behavioral patterns are created on log solutions such as NetFlow Monitors, SIEMs, and IDS/IPSs. For mitigating controls network and host solutions such as NetFlow enforcement solutions, EDR, and ACLs and network equipment such as firewalls, proxies, and routers need to be considered with implementation tasked out to the appropriated departments or teams. All remediation efforts need to be thoroughly documented in the technical exploit management database and tracked. If a patch is applied the compensating controls can be evaluated and possibly removed with an update to the technical exploit management database. No exploits found in the data base should have and "accepted risk" rating. The reason for this is due to the nature of that rating. When items become "accepted risks" they become forgotten risks and lead to a complex set of innumerable low-level risks that usually constitute and equally innumerable series of highly risky exploit chains. For the lowest risk technical exploits it may be best to create alerts for those a variety of alerting tactics can be used to rule out false positives. The rules for low risk items should be fairly precise to reduce false positives or should have a high threshold (or silently audit) for items that have a large false positive rate. Either way, very low risk technical exploits that are applicable to the environment should be logged and occasionally reviewed for signs of active exploit.

Most organizations are likely to already have a technical exploit management program; however, those organizations may not realize such and may have embedded it in other initiatives like patching and network security. To ensure a unified understanding of what constitutes a technical exploit management program a definition was provided along with the differences between known and unknown technical exploits. The attention was then turned to discovery of applicable technical exploits. Technical exploits require response actions. A brief discussion of one way to practically implement a technical exploit management program was provided as a finally point for consideration.

Thank you for reading

-Andrew Kosakowski

www.Anthko.com