Elton Wong

Homework – IR & ATT&CK

What key evolutions happened with this campaign in its 4 main iterations?

At the very first iteration of this campaign, the actor relied heavily on the filename to lure the target to click on the spearphishing attachment (user execution) and execute the malicious code. In the second iteration, the spearphishing attack evolved into an email with LNK shortcut which meant to run a series of PowerShell scripts to extract a payload from the shortcut to install and execute the malicious code. Also, the defense evasion has advanced from traditional modify registry to deobfuscate/decode files or information. The third iteration switches the focus of the attack into using Flash Player exploit (Dealers Choice) and using the custom cryptographic protocol for command & control. In the last iteration of the campaign, the actor uses more variety of attack techniques such as logon scripts and hidden files to maintain its accessibility to the target machine, privilege escalation(process injection), and collection (clipboard data).

What datasets/data feeds would we need to have coming into the SIEM to detect SOFACY?

- Execution/defense evasion/discovery
    - command-line arguments and its analytics
    - execution file paths
    - process
- Initial access
    - network intrusion detection system
    - email gateway
- Persistence
    - registry values
    - running process for action
- Privilege escalation
    - Windows API calls
    - DLL/PE file events
    - pipe creation and connection events
- Command and control
    - network data for uncommon data flows

Would we use static correlation or user/entity behavior analytics, or both?

In this scenario, I would use static content to create the best result for any detecting security event. I believe there is a value of deploying user behavior analysis as well but it seems like it won't be a great option due to the lack of information.

For the static content, I will use the Top-Down Bottom-Up Middle-Out approach for my use case modeling. In the top-down view of the SIEM system, the SIEM solution would be at the root node and it links to the systems node, the specific version of systems, and its specific types of logs. For example in the use case of detecting spearphishing emails, the SIEM solution would connect to the email system

and gather its system log data into the solution, which include email topics, attachments, the content of the email, sender address and timestamp. In the bottom-up view, data should be categorized and reduce to its minimum functional form to ensure the efficiency of the SIEM system, but of course, it would be done based on not sacrificing any functionality. In the part of middle-out of the SIEM design process, we need to construct use cases from the data to support its objective. In the case of detecting spearphishing emails, we can ideally construct a machine learning model to check if the inbounding emails contain any malicious content and hint the users to take action against those activities.

Thinking proactively, if we had some level of confidence that SOFACY was active in our environment, list and prioritise (triage) what incident response activities we would want to carry out?

- Detect the malicious activity
- Contain the infected environment
- Eradicate any malicious code in that environment or reset the whole system
- System Recovery
- Preparation and continuous improvement of incident response

How could we work ahead of the adversary? List some specific technical controls you would work with the IT/tech teams to implement in order to prevent key SOFACY techniques.

- Ensure a secure antivirus profile is applied to all relevant security policies, configured to block on all spyware severity levels, categories, and threats, and set to block on all decoders except 'imap' and 'pop3'
    - Spearphising emails, user executions, standard application layer protocol
- *Ensure DNS sinkholing is configured on all anti-spyware profiles in use*
    - User executions, standard application layer protocol
- Employ heuristic-based malware detection
    - Software packing
- Install endpoint detection and response (EDR) product
    - Spearphishing emails, user executions
- Email dentonation applicance
    - Spearphishing emails
- Monitor scheduled tasks, autorun processes and command-line arguments
    - Not sure if the security team can collect those data
- Implement early warning system for detecting intrusion activities