Moving Target Defense – a Journey from Idea to Product

Jason Li
Intelligent Automation Inc.
15400 Calhoun Drive, Suite 190
Rockville, MD 20855
1-301-294-5275
jli@i-a-i.com

Justin Yackoski
Cryptonite, LLC
15400 Calhoun Drive, Suite 190
Rockville, MD 20855
1-301-294-4251
jyackoski@cryptonitenxt.com

Nicholas Evancich
Intelligent Automation Inc.
15400 Calhoun Drive, Suite 190
Rockville, MD 20855
1-301-294-4254
nevancich@i-a-i.com

ABSTRACT

In today's enterprise networks, there are many ways for a determined attacker to obtain a foothold, bypass current protection technologies, and attack the intended target. Over several years we have developed the Self-shielding Dynamic Network Architecture (SDNA) technology, which prevents an attacker from targeting, entering, or spreading through an enterprise network by adding dynamics that present a changing view of the network over space and time. SDNA was developed with the support of government sponsored research and development and corporate internal resources. The SDNA technology was purchased by Cryptonite, LLC in 2015 and has been developed into a robust product offering called Cryptonite NXT. In this paper, we describe the journey and lessons learned along the course of feasibility demonstration, technology development, security testing, productization, and deployment in a production network.

CCS Concepts

• Security and Privacy → Network security; System security • Networks → Network components.

Keywords

Moving target defense; enterprise network security; IPv6

1. INTRODUCTION

Today's enterprise networks are "sitting ducks" waiting for attackers to exploit them. TCP/IP networks were not designed with security in mind, and many widely used protocols have fundamental design flaws that still have not been adequately addressed. Determined, sophisticated attackers have a variety of tools at their disposal to get inside an enterprise network, bypass any current protection technologies, and attack the intended targets. It is unfortunately a given that attackers will get into any network, and the only question is how much damage will be done before the attack is discovered.

In current practice, detection systems check against signatures of known attacks but do not protect against unknown attacks. Firewalls are good at stopping attacks from entering the network but there is no protection once the attacker gets past them. Basic randomization techniques modify aspects of the enterprise network to improve resilience, but do not contain an attack once it has started and do not prevent against misuse of credentials. Honeypots and honeynets create fake targets to distract attackers, but do not provide any protection once the attacker correctly finds the real target. None of these technologies stops the now common practice of attacking the enterprise network from within using zero-day exploits, stolen credentials, and other sophisticated tactics. An innovation in cyber security technology is needed that goes beyond what the current state of the art has to offer.

As part of the research and development (R&D) community, the authors have participated in developing innovative technologies that provide moving target defense (MTD) capabilities to enterprise networks [2][4][5][6]. The initial Self-shielding Dynamic Network Architecture (SDNA) technology was purchased by Cryptonite, LLC in 2015 and has been developed into a robust product offering called Cryptonite NXT. Cryptonite NXT dynamically alters an enterprise network's appearance and behavior to stop cyber-attacks, including zero-day and targeted advanced persistent threats (APT), while maintaining transparency to the user, application, and operating system. Cryptonite NXT prevents an attacker from targeting, entering, or spreading through an enterprise network by adding dynamics that present a changing view of the network over space and time. Cryptonite NXT uses IPv6-based cryptographically strong dynamics to alter the appearance and behavior of each host in the network while remaining transparent to the user and backwards-compatible with IPv4. Cryptonite NXT prevents malicious packets from even reaching the network hosts no matter how much time or resources the attacker spends, thus keeping vulnerable systems safe. If an attacker gains a foothold inside the enterprise network, for example, a malicious insider or a host compromised by a phishing attack, Cryptonite NXT limits the attacker's ability to spread and operate by constraining each host to an abstract, modified, and obfuscated view of the network. Cryptonite NXT consists of control software and network appliances that together establish a security infrastructure to manage the flow of network traffic without impacting usability.

Cryptonite NXT is a unique offering in that it is pro-active in its protection and that it does not depend on continuous intervention from the Information Technology (IT) department to maintain a secure environment. It is anticipated that the Cryptonite NXT product will greatly enhance the cyber security posture and reduce cyber security associated damage costs in a wide range of commercial and government sectors.

In this paper, we briefly describe the Cryptonite NXT technology, followed by lessons learned during the progression to technology development, security testing, productization, and deployment in a production network. The purpose is not to report a particular technology; instead it is meant to share our experience of one MTD technology from its inception to product.

MTD'16, October 24 2016, Vienna, Austria
© 2016 ACM. ISBN 978-1-4503-4570-5/16/10…$15.00
DOI: http://dx.doi.org/10.1145/2995272.2995286

2. Cryptonite NXT TECHNOLOGY

Cryptonite NXT[1] uses packet manipulation, policies and rules to manage the competing goals of simultaneously securing the network while providing authorized users access to needed services. Several appliances are deployed in a protected network such that Cryptonite NXT logically sits between each protected device (e.g., user workstations) and the network, adding and removing dynamics, creating abstraction, and applying controls as traffic enters and leaves the devices. We use the term *endpoint* to describe a device protected by Cryptonite NXT.

The resulting view from the perspectives of the user, application, operating system (OS), and network is illustrated in Figure 1 (included at the end of the paper). When in the network "below" Cryptonite NXT, packets appear to be traversing a dynamic network. On the operating system, application, and user side, Cryptonite NXT creates a semantically valid view of the network to ensure compatibility, while at the same time retaining some dynamics for security. Through cryptographically-secure mechanisms, dynamics are added to the network to prevent the attacker from gathering and acting on information about the network.

This semantically correct view is presented to the OS via the use of Token IPs. Token IPs are temporarily active IP addresses that are provided by Cryptonite NXT to the OS via domain name service (DNS). Cryptonite NXT transparently maps Token IPs into and out of the network, allowing Cryptonite NXT to both conceal the actual network structure as well as to change the underlying operation of the network without the operating system's knowledge. The Token IPs are only valid for use by the particular OS they are provided to. By integrating dynamics into a common architecture, the protection provided by multiple mechanisms can be constructively combined to enhance overall security. The set of dynamics used by Cryptonite NXT are as follows.

- **Dynamic Token IP Mapping:** Cryptonite NXT transparently rewrites packets entering and exiting the endpoint, inserting and removing the Token IPs. This mapping prevents the endpoint and any malicious activity running on it from observing or using real addresses in the network as well as from determining the network topology or services.
- **Dynamic Bits:** While traveling within the network, the valid source and destination IPs change on a per-packet basis. A subset of the IPv6 addresses is assigned using information known only to the Cryptonite NXT appliances involved in each network connection. These bits prevent a non-protected device from communicating directly with protected endpoints or escalating privileges.
- **Intermediate Node Redirection:** To provide additional security, packets travel through one or more *Intermediate Nodes* before reaching their final destination. Intermediate Nodes allow Cryptonite NXT to transparently redirect the path of traffic through the network and conceal communication patterns by manipulating the IP headers of each packet.
- **Dynamic DNS Mapping:** All DNS responses passing through the network are mapped into the Cryptonite NXT dynamics and then replaced with a semantically valid response. This mapping is used to transparently provide Token IPs to the endpoint based on policy.
- **Service-level Spatial Dynamics:** As constrained by policy, Cryptonite NXT dynamically alters the meaning of each Token IP at a per-user, per-service level. Unless currently required for use by a given user, communication is redirected to a honeypot, black hole, or other harmless destination. Since each user and thus each host has different needs, each endpoint is effectively given a different and varying view of the network. This capability limits attack spread within the network and complicates network mapping by attackers.
- **Network-wide Security Posture:** The system can be pre-configured with multiple policy sets. At any given time, only one policy set can be actively applied to the protected network. The policy set dictates the legitimate connections on each user, and endpoint. Administrators can switch between policy sets to dynamically adjust each host's view of the network in order to gauge a suspicious host's reaction or to quickly impose a higher security posture in response to a perceived threat.

Each Cryptonite NXT appliance uses standard security certificates and authentication mechanisms to independently verify the source of each packet and connection. The network's behavior and appearance can then be varied based on the result of this verification. The availability of each service can be changed in response to the proper credentials with a provable audit trail, giving administrators the ability to enact highly restrictive but flexible security policies. As a result, endpoints are unable to misuse the credentials supplied by the user or connect to arbitrary destinations without the user's knowledge.

Equipped with these dynamics, Cryptonite NXT possesses the following capabilities:

- **Security by default:** Cryptonite NXT provides strong protection on a per packet basis against undesired network actions from external and internal attackers. Dynamic views of the network are created based on defined policies, matching network availability with a user's "need-to-know". Packets are only delivered through the network if they are verified to match the dynamics of the network.
- **Limited information gathering and protection against attack spread**: Cryptonite NXT prevents the endpoints from knowing or spoofing each other's identity, effectively conceals the structure of the network and limits discovery of accessible services. In addition, Cryptonite NXT exercises fine-grained, per-user/per-service control of network dynamics, preventing observable information from being used to conduct an attack efficiently.
- **Compatibility and transparency:** Cryptonite NXT supports existing network hardware, operating systems, and applications. Cryptonite NXT provides security while maintaining complete transparency from the perspective of the user, application, and operating system. The OS and higher layers do not need to be modified to gain the advantages provided by Cryptonite NXT.
- **Fault-tolerant, distributed operation:** Cryptonite NXT's architecture does not cause the network to be fragile or add new potential points of failure. The decentralized design provides fault tolerance and minimal control overhead.

[1] The term *SDNA* refers to the original research technology, and the term *Cryptonite NXT* refers to the current technology and product. Cryptonite NXT is a trademark of Cryptonite, LLC.

It is noted that the goal of the above section is to provide a brief summary of the subject technology to lay out the foundation for discussions about the journey and lessons learned. Specific technical details of the Cryptonite NXT technology can be found in previous publications [7][8].

3. TECHNOLOGY DEVELOPMENT

In this section, we describe the lessons learned during the process of technology development, mostly from the viewpoint of industry working on applied research and development.

3.1 Feasibility Demonstration

In the applied research community, especially among those involved in developing innovative technologies for identified problems with immediate, practical motivations and implications, it has become a common practice to execute feasibility demonstrations in relevant environments using realistic scenarios. A feasibility demonstration provides an opportunity for the researchers to showcase the envisioned capabilities, identify potential technical risks, and obtain feedback from the sponsoring customers. Achieving an appropriate level of realism is always a challenge. The time frame for such feasibility demonstration could span from several months to a year depending on the particular research program.

In order to succeed in such a demonstration event, it is typical for the researchers to plan the work to be done in some backward manner: define what and how to demonstrate; identify what needs to be accomplished; make a work plan and allocate resources; and execute the work plan rigorously. In many cases, a smaller number of (elite) team members is gathered, and a mindset of fast prototyping toward a very specific goal is needed. However, fast prototyping should not imply ad hoc practices. On the contrary, the team usually adopts some agile development and test process, with short-interval sprints and weekly (or even daily) team stand-ups being commonly exercised. In most cases, due to limited time and budget resources, the team leverages current capabilities and seeks open-source tools that can fit the program and demonstration needs. However, it is important to be aware that previous capabilities should not dictate the current program direction (i.e., not falling into the trap of "if all you have is a hammer, everything looks like a nail"), and also that open-source tools were not created particularly for the work at hand and may present licensing issues.

In particular, during our course of feasibility demonstration, we found it prudent to 1) engage the customer early to agree upon scenarios and expectations; 2) create and approve a feature list for implementation; 3) prioritize the order of implementation from the feature list; 4) exercise regular stand-up meetings to assess progress and stay focused; 5) execute tests frequently and incrementally with regression; and 6) document tools leveraged, their limitations (both technical and non-technical such as Intellectual Property rights), and plans to overcome such limitations for the next phase of development. By focusing the demonstrations on specific aspects of the problem and keeping realistic assumptions in mind, the technology used in the demonstrations always maintains a viable exit path for productization.

3.2 Product Development

Once the feasibility demonstration is successful, giving confidence for future success, the next phase of development needs to focus on product development. Ideally, the initial technology was already created with the goal of technology transfer to minimize this effort.

There are some important changes that we observed during this phase of development. The first change was in mindset. Focusing on R&D goals and fast prototyping, the mindset was "let's put together something real cool real fast to show that our solution could work". For product development, the new mindset needs to be "now this solution has to actually work well 24/7 in the face of a variety of unexpected situations". This mindset change is not minor, since it means a change from research to engineering approach in work and possibly changes in the team members themselves. It is not easy to switch out former team members especially when the program now has more work and a future. But sometimes it is a hard decision that must be made. The team also tends to become larger with new members, with a mixed background in research, software engineering, and different ways and attitudes of working.

The next change is to introduce more rigorous systems engineering, for example, as embodied in some form of software development life cycle process. Program management becomes essential, and dedicated staff with pertinent experience is needed. Requirements, schedules, and work breakdown structures need to be identified and executed. Knowledge transfer among team members and rigorous team stand-up meetings become critical to manage progress, identify and mitigate technical and non-technical risks.

It is worth emphasizing the role of testing during the phase of product development. In general, we adopted the well-known principles of systems engineering and executed incremental testing (with necessary regression tests when major revisions are made) at component, sub-system, and system levels against the different levels of requirements. Specifically, we learned some interesting lessons: 1) Testing should not be underestimated – just because something should work does not mean it will work – and that is a major differentiator from prototype to product; 2) Testing should not replace design, and careful design with good development practice pre-determines what the component/system/product can do; 3) Testing takes discipline – some researchers may not like it – so it is important to identify people with determination for this job; 4) Informed testing can bear more benefits in a security product such as Cryptonite NXT, and an informed tester can go further in terms of functionality and expected behaviors; 5) It is important to know what to test for and how to test it; the importance of test planning cannot be overstressed; and it is surprising how many times the tests are not thought through; 6) Test small and test often without waiting until it is too late; do not forget the regression and integration tests; and lastly, 7) For a security product, establish a testbed and perform tests on it.

4. SECURITY ANALYSIS AND TESTING

As part of our development effort, we also executed security analysis to assess the solution and established testbeds for capability demonstrations.

4.1 Security Analysis

Cryptonite NXT uses a combination of techniques to prevent an attacker from both observing and acting on information about the network. Since 100% accurate detection remains elusive in the field of network security, the goal of Cryptonite NXT's dynamics is to 1) keep the attack in the reconnaissance phase as long as possible, and 2) make the execution of many attacks infeasible due to the risks and logistics involved.

The analysis below begins by assuming that an attacker external to the Cryptonite NXT-protected enclave attempts to scan or attack a host within the enclave. Assuming the attacker somehow identifies or guesses the address of a given endpoint in the Cryptonite NXT-protected enclave, the attacker cannot simply connect to that address. This restriction exists because Cryptonite NXT uses IPv6 to allocate each protected host 2^{40} (about 1 trillion) possible addresses, only one of which is valid for a given packet. The valid addresses used for both the source and destination of a packet (totaling 2^{80} combinations) are calculated using a hash of the packet's metadata and a temporary key, meaning that the correct addresses for each packet are different. Since an attack originating from a non-Cryptonite NXT host cannot obtain the key used by Cryptonite NXT to compute the correct addresses, the attacker must guess the correct hash for a current temporary key. Each guess can only be validated by actually sending a packet containing the guessed data, requiring the attacker to expend significant effort. Any such guesses are identifiable using a simple hash calculation, allowing such traffic to be immediately discarded and a reliable notification of the suspicious behavior generated. This notification could be provided to an intrusion detection system (IDS) or other monitoring software.

Next, consider a generic attacker operating from within the Cryptonite NXT-protected enclave. There are two possible variations. First, the attacker may be present at a device in the network that is not Cryptonite NXT-protected (e.g., a router, legacy device, etc.). In this case, the attacker cannot communicate with any Cryptonite NXT-protected hosts for the reasons described above. Second, it is possible for an attacker to compromise a Cryptonite NXT-protected host and attempt to use that as a launching point for further attacks, for example by exploiting a vulnerability within the OS. In order to avoid requiring modifications to the OS and any applications, Cryptonite NXT must provide the endpoint with a semantically correct view of the network that masks the network dynamics. This same view is presented to both malicious and benign applications on the endpoint as Cryptonite NXT cannot determine the intent of a particular application or packet. Further, any actions taken by Cryptonite NXT must not prevent a legitimate user from performing legitimate tasks (e.g., checking email, transferring files, etc.).

To address potential misuse in this scenario, the view provided by Cryptonite NXT to the endpoint is an abstracted view with dynamic availability/visibility to the rest of the network. In other words, only the minimum necessary set of correct semantics is provided to the endpoint on a "need-to-know" basis, providing spatial dynamics that create different policy-defined views of the network from each host's perspective. This approach allows Cryptonite NXT to conceal the true structure of the network and the locations of services, provide automatic fine-grained dynamic availability at the per-user and per-service level, and prevent direct access to the network.

Cryptonite NXT further improves security in this scenario by integrating with the user's credentials and authentication. This integration reduces a compromised endpoint's ability to act as a man-in-the-middle in order to misuse the network.

Finally, Cryptonite NXT can effectively contain the impact of an attack after it has infected a host and tries to spread to the rest of the network. A worm was used as a concrete example illustrative of many of the actions taken by various types of attacks. Please see [9] for detailed descriptions.

4.2 Establishing a Testbed

The technology should be evaluated from different perspectives, such as security, performance, stability, and usability. Some parameters, such as performance, can be accurately evaluated only in a noise-free environment where traffic is deterministic at all times. On the other hand, parameters such as security should be evaluated in an isolated environment (to prevent contamination of the hosting network) where noisy traffic is recommended. To this end, test networks need to be established and maintained with complete control over the devices, their configurations and traffic flowing across them with the authority to reconfigure them as needed for various tests. In general, establishing a test network is an iterative process involving four phases:

1) Network design: Based on the purposes and requirements for the tests at hands, the network design starts with the overall network topology, network assets configuration planning, and profile of application services and usage patterns. Critical aspects of a typical operational environment such as subnet addressing scheme, access mechanisms (direct access, VPN, etc.), authentication mechanisms and network management tools should be designed. It is important to keep this design as close to a typical target site network as possible. It is also necessary to design it in such a way that the network as a whole or sections of it can be switched between protected and unprotected on-demand so that testing and analysis can switch between the two with as little downtime as possible.
2) Test development and evaluation methodology: This phase determines the services, number of servers, user types, users and traffic generators to be deployed in the network. The network and test design also includes the data collection tools and mechanisms at the appropriate levels of the protocol stack (link, network, transport, or application layer) necessary to collect the data required for evaluation. This test suite can be a combination of written instructions, automatic tests and their test harnesses, scripts, traffic generators and data collection mechanisms.
3) Base network installation: The test network is first established as a traditional network with typical configurations per the design. IPv6 and IPv4 may be enabled along with all supporting capabilities such as routing, DNS, and DHCP. The services and tools necessary for network operations and management are launched and basic data collection mechanisms such as data taps at the routers and switches are implemented. The access mechanism and authentication systems are introduced before applications and users are added. Finally, the servers, services, users and traffic generators are installed and configured.
4) Migration: Migrating the test network involves inserting the security technology in the network. In the Cryptonite NXT case, appliances are added to the network topology. The outcome of this process is that all network traffic first passes through Cryptonite NXT appliances where dynamics are applied and access control policies are enforced at the packet level. This ensures that Cryptonite NXT processes all packets before they are forwarded to the protected OS and applications. Configurations are updated as needed to finish constructing the network.

4.3 Evaluation Metrics

Test network establishment and test development go hand-in-hand with the development of evaluation metrics to highlight key aspects of the Cryptonite NXT technology, including performance, security, and usability.

The following host- and network-based performance metrics are applicable: 1) *Throughput* refers to the rate of successful packet delivery over the network in bits (or bytes) per second. This can be measured by transferring a large file, and counting successfully transferred packets in a specified time window. 2) *Latency* refers to the amount of time it takes for a packet to travel from one point to another in the network. For example, round trip time measurements via *ping* can be used to assess latency between two *ping* endpoints. *Jitter*, variation in latency, may also be relevant for some applications. In addition, *response time* can be measured for a particular user application, such as email access or web browsing. 3) *Packet loss (rate)* refers to the number (or percentage) of packets travelling across the network without reaching final destination.

Security metrics must help answer key security questions: is the enterprise more secure than before, and is the enterprise secure enough? The initial set of security related metrics are identified as follows: 1) *Discovery susceptibility* refers to the amount of host and network information that can be revealed by typical network scanning tools, such as *nmap* and its variants. 2) *Resistance to known attacks* refers to whether or not known attacks can be successful. For example, does a specific attack from a common attack tool still work or work with minor changes? 3) *Attacker workload* refers to how much effort (e.g., probing packets, failed attempts) and risk (e.g., the likelihood of actions or mistakes allowing detection) the attacker may have to render in order to gain (sufficient) access into a target network. 4) *Spread containment* refers to how widely an infection can spread to the target network, without and with protection. 5) *Access control* refers to whether or not illegal information flow is correctly blocked, or conversely whether any legitimate traffic is incorrectly blocked, per the current access control policy enforcement. 6) *Malicious data resistance* refers to whether or not malicious data, either from outside or inside of the enclave, can traverse the network as intended. For example, can the system detect, identify, and drop/redirect such traffic reliably and efficiently?

Note that while effective for basic measurements and demonstrations, the problem with all known security metrics is that the connection from these values to the impact on real world attacker tactics is weak. For that reason, the security evaluation should not be viewed as complete until an experienced (and ideally professional) penetration tester has fully understood the technology and thrown their bag of tricks and experience at attacking the protected network. Such penetration testing is discussed in more detail in Section 6.2.

Finally, usability must focus on common applications, use cases, as well as IT department feedback. Usability metrics, from the perspective of the *users*, mainly focus on user transparency with common applications and use cases, such as email access, web browsing, and file transfer. Objective metrics, such as file transfer rate and email response time, should be measured, and feedback from actual user experience should be collected and analyzed. Usability metrics, from the perspective of the *administrators and IT staff*, are different. These indicate the ease of managing, monitoring, and maintaining the network. They are more qualitative than quantitative and are measured through targeted surveys and spot checks. One measurable metric is the availability of the network, which is the percentage of time the network is running without any problems impacting availability of services.

Using such methodology, different test networks were established with different topology, device configuration, application profiles, usage patterns as well as various attack vectors and evaluation metrics. Extensive testing was executed with results and analysis, and previous publications reported some of these results [7][8]. Next we briefly describe a recent testbed demonstration scenario.

4.4 Testbed Demonstration and Scenario

The particular demonstration scenario was based on the Target Stores, Inc. data breach, which is publicly available and relevant to show Cryptonite NXT's security capability.

In this scenario, an attacker placed malware on the laptop of an unaware HVAC service technician. This laptop was then directly connected to the internal network to perform needed HVAC service, bypassing the firewall. This allowed the malware to compromise the billing server which was used as the entry point for the rest of the attack. From the billing server, the software update server was compromised, as were several other servers. A malicious software update was placed on the update sever. When the point of sale (POS) terminals performed their next automatic update, they became infected and began capturing credit card numbers. These numbers were sent to a dump server, another host in the network that the attacker had compromised. Once a sufficient amount of stolen credit card data was collected, they were transferred to another compromised server with outside network access and copied through the firewall to a drop site under control of the attacker.

We have mapped this breach into a testbed. The attack consists of 7 different stages (see Figure 2 and Table 1 at the end of the paper) and we determine the benefit of Cryptonite NXT by running the attack with and without Cryptonite NXT. Overall, Cryptonite NXT stops 5 of the 7 stages of the attack. Only a single stage needs to be stopped for the attack to be unsuccessful. The attacker is stopped in 5 different stages by Cryptonite NXT, allowing a considerable security margin. Several of these, in particular stage 2 and 3 are applicable to almost all networks. For stage 2, no unprotected devices can communicate with protected devices due to Cryptonite NXT's dynamics. For stage 3, network scanning tools are shown a Cryptonite NXT-created abstract view of the network that limits misuse of the information for an attack. The other stages fail because the attacker can only use communications that are stopped by Cryptonite NXT. Note that some details are intentionally omitted here for various security reasons.

For comparison, a FireEye protected network would only generate generic warnings, but not stop stages 3 and 4 (see Table 1). In fact, FireEye was running in Target's network when it was successfully attacked, and these alerts did not stop the breach. With that said, FireEye still provides a very worthwhile technology to deploy. Today, FireEye would undoubtedly generate very specific high priority warnings if the same malware was used and so clearly some protection against *known* attacks is provided.

This comparison demonstrates the difficulty in quantifying security benefits. Failure to detect today does not mean failure to detect tomorrow as new malware signatures are added. Similarly, Cryptonite NXT excels at stopping and containing both known and unknown attacks, while FireEye is designed for detecting known attacks. These are different technologies with different roles and impacts on a per-attack and per-organization basis. FireEye will certainly detect some known attacks in time to stop them. Looking back to the research phase, the difference between building the technology on the dangerous assumption that the attacker will only use a known attack versus our assumption that the attacker can and will do anything leads to clear implications in the real world. Assumptions should thus be as realistic as possible and their implications should be well understood as the technology grows.

5. PRODUCTIZATION

With the great potential evidenced in numerous demonstrations, we have taken the endeavor to fully commercialize the SDNA technology, which was purchased by Cryptonite LLC to focus on product development, launch, marketing, and sales. Since its inception, we have encountered some interesting experiences not typically seen in the R&D community. This section describes our adventures, which could be helpful for others in taking security research to the productization stage.

5.1 From Technology to Market

After the decision was made to bring the technology to market, the focus rapidly shifted from expanding on the science behind the technology to creating a product that can incorporate the best aspects of the underlying science in a package that could be easily approached by users. Specifically, the definition of “better” changed dramatically. Showing that the technology would achieve some level (say percentage) of improvement over another technology if both technologies were productized becomes meaningless. This observation is not because accurately quantifying the security benefits of any technology remains elusive in general, but because the question is instead whether the product provides significant customer benefits and is at the same time no harder or more expensive to deploy and maintain. Products are measured using actual data instead of theory.

The pros and cons of the technology also become viewed in a different light when moving to a product. An edge case no longer matters if it cannot happen in the real world, perhaps due to some implementation subtleties. Similarly, some edge cases may turn out to be the normal or most common case due to incorrect assumptions. A benefit that is provided by the technology only matters to the extent it can be realized by the customer. For example, substantial protection offered for IPv6 traffic is not a selling point for the majority of customers who are currently using IPv4. Protection for both is needed. Along the same lines, the biggest weakness of a product is not defined by the science behind it but by whatever aspect of the product most strongly conflicts with the customer’s expectations. The customer may be unwilling to take a certain action for purely non-technical (but completely logical) reasons that cannot be assumed or theorized away.

All of the assumptions made during the technology’s design are no long valid when a product is placed in the real world, no matter how much the peer reviewers, during the publication review process, agreed that those assumptions were in fact valid. Previously minor details of how to authenticate a user, upgrading the software version, etc. must all be fully solved. A product, especially one that is providing vital protection or functionality within a customer’s network, needs to operate 24/7 without any major issues and with an easy mechanism to efficiently resolve any minor issues. Unexpected events will happen. The easiest are when users forget their password at 3am. An overnight Microsoft update might alter something that requires users to take a different step when authenticating to the network, making it hard for the users to read the emergency email providing the new instructions. Hardware, power supplies, and Ethernet cables will fail and when they do the product needs to make it clear to the user where the problem lies.

It may be obvious that an extensive market study is needed, which should include both the literature survey and evaluation of the current and forthcoming products on the market. The former is something most researchers are familiar with. The latter, however, poses some interesting questions, and sometimes the answers need to be provided from unfamiliar angles so that they resonate well with the different audience. The former also prizes new, unexplored directions while the latter sees unproven markets as high risk.

The example set of questions include: “What does your product really do?”, “What are some other products out there?”, “Why is your product better?”, “Why should we buy your product?”, etc. The typical academic way of answering these questions, such as “previous works fail to do many things, and our work overcomes all those drawbacks, and here are our graphs to show the evidence”, will almost always fail in these situations. The reason is simple – these are not the answers that were sought after by a product-oriented audience. The right answers need to directly address issues related to the product’s *real capability*, how it can work with other installation in the enterprise, and how the product may impact the enterprise operation, just to name a few. To evaluate other products, in addition to pointing out what they may not do, it is more prudent to understand that competition is probably good for the overall market needs, and being able to work with other products will only increase the chance that the product gets adopted.

Though there may be justifiable confidence in the product, it is critical to emphasize that, as it is brought to market, a technology that fully solves all cyber security problems is unlikely to ever make it to market. The sheer number of minor details that needs to be handled to cover every aspect in a customer friendly way makes it impossible. More importantly, a customer already has a variety of products in their networks, perhaps too many. Claiming to be a better version of widget A is easily understood at many levels. Claiming to be a better versions of widgets A, B, and C is also understood but is often met with a justifiable and somewhat exaggerated level of skepticism. This is not an insult to the technology; it is just a customer having a natural reaction to what is either too radical and risky, or a change to their current business, or something that must be too good to be true. Claiming to be a completely new widget D that has no direct comparison requires the most amount of explanation and convincing since it requires a fundamental alteration of the customer’s philosophy in some ways. Overall, it is best to provide a small, well rounded, usable, and simple solution that does an extremely good job at fixing just a handful of very critical problems a customer has.

We undertook a series of validations of the product to understand all of the real world scenarios not considered during the initial technology phase, reduce the perceived risks involved with deploying the product, and prove the benefits of the technology. Fortunately, the vast majority of the issues found were already considered early on in technology development, but this was still a critical step in turning the technology into a marketable product. If nothing else, no one believes something works if it can only be seen on paper.

Moreover, during the process of selling a product, the questions of “Who are your current customers” or “Where are your products currently installed” were always asked. Contrary to the Internet bubble days, initial deployment successes are required before a technology or product is taken seriously.

Finally, inserting innovative cyber technologies into existing market ecosystems is significantly more challenging than simply packaging software and selling it. As mentioned above, different skill sets, discipline, and even mindsets are needed for which technical innovators may not have any formal training or prior experience. For more information along these lines, readers are encouraged to find more details in the invited talk given by Gordon during Usenix Security Symposium 2015 [1].

5.2 Functional Verification and Validation

As part of the product development and pilot deployment process, a test network was established and the test suite was developed to perform functional verification and validation testing. The test suite was also iteratively expanded and improved throughout the entire process. As a second step, the product was deployed in actual customer networks to provide real-world testing and to understand how well the system could be managed in daily use.

In addition to (the obvious) testing of correct operation of the applications and usage patterns, we routinely executed a variety of tests, which include:

- IPv4 and IPv6 address assignment and renewal via DHCP;
- Dynamic route discovery;
- DNS name resolution, internal and external;
- ping and traceroute between endpoints;
- Windows network login, Unix-based network login;
- Networked file sharing and resource discovery;
- Applications using protocols such as http, ssh, smtp (email);
- Audio and video streaming over UDP;
- Large file downloads and uploads;
- Networked printing;
- SNMP-based network data collection.

Each product release, and in particular the product to be used in any deployment, needs to pass all such tests before its release and subsequent deployment.

5.3 Security Verification and Validation

Similarly, we established a test environment to cover comprehensive test cases including network discovery, scanning, accessing services, spoofing, and initiating malicious traffic. The test suite during our internal testing is summarized as follows:

- **Discovery and scanning from outside:** This test case evaluates the ability of unauthenticated hosts from outside the protected enclave to map the target network and the underlying services available via IPv6.
- **Discovery and scanning from an endpoint:** This test case evaluates the ability of a (compromised) endpoint from inside the enclave to map the network.
- **Discovery and scanning from a non-protected device:** This test case evaluates the ability of a non-protected host (e.g., a rogue device added to the network) inside the enclave to map the current network.
- **Malicious data between endpoints:** This test case evaluates Cryptonite NXT's ability to mitigate malicious data from passing through the network between endpoints.

5.4 Performance, Performance, Performance

When you ask some customers what they are willing to give up in order to improve their security, you may get one of two answers. Some may already have given up a significant portion of each endpoint's CPU and memory for virus scanning and other endpoint protection software, in which case they will only add products anywhere in their networks if the product reduces the amount that the customer has already given up. In other words, your product has to make up for the shortcoming of others. Other customers may likely respond that they are not willing to give up anything for security. But, given the currently well-publicized risks associated with insufficient security, why are people still not willing to give up something for better security? Try adding a two second delay to each of the possibly dozens of DNS query your browser makes on every page load and see if you still feel that way. Alternatively, just insert your chip and pin enabled credit card into the nearest credit card reader and you will unconsciously begin to curse whoever failed to fix a "minor" per-transaction delay.

Cryptonite NXT has the potential to most noticeably affect users in terms of performance, specifically throughput and latency. Early on during technology development, it was sufficient to show almost any throughput numbers with the caveat that "in theory" only so many CPU cycles were needed to process each packet. Similarly, some latency numbers were shown and latency would improve "in theory" if more processing and memory power can be consumed. This is the familiar tone we all use during research and development. However, as Cryptonite NXT moved to a product, the actual observed performance was the only thing that mattered.

By at least an order of magnitude, the single largest engineering change in creating the product was due to performance. We needed to achieve at least Gigabit wire speeds and were able to achieve only about an eighth of that in a small, cost-effective form factor. The alternative was a quad-core, high-end server CPU that was simply not a scalable or marketable solution.

It was eventually realized that most hardware and software is not designed in a manner that is compatible with processing traffic without impacting performance. How can this be? It turns out that the major performance impact in a typical network is as traffic enters a device and is handled by the operating system, drivers, hardware, etc., which combine to create a not normally perceptible, but still non-trivial performance impact. This makes sense as there is no need to optimize beyond that. However, we need to send and receive within our own separate device, plus the protected device still needs to receive, effectively making the performance 3 times worse than normal.

In the end, the underlying software was ported onto a specialized, reasonably priced MIPS-based network packet processing CPU that provides expedited packet and cryptographic processing speeds. By leveraging this hardware acceleration and completely bypassing any operating system or drivers (similar in spirit to how *zmap* outperforms *nmap* for Internet host enumeration [3]), Cryptonite NXT now achieves Gigabit and faster throughputs using a fraction of the available CPU resources with latencies measured in tens of microseconds. Resolving the performance issues was the only viable path forward to achieve a successful product.

In retrospect, if the performance numbers had been seriously looked at during research and development instead of being solved "in theory", the architectural switch could have been made much more easily years ago. The mistake was made because we (as researchers) simply were not interested in actually solving this "theoretically solved" problem (we had many others to solve), and even if we were interested it is likely that research-oriented funding sources would be justifiably uninterested in funding what are seen as engineering pursuits. In the end the architecture change was needed and beneficial for a variety of reasons well beyond performance. The R&D platforms that were originally used were not designed for product use.

Once it becomes clear that a given hurdle must be overcome, a choice has to be made. You can accumulate enough hurdles that the technology becomes impractical and ends up on a shelf as some interesting reading material. You can always hope to get lucky that no other hurdles come up and that one day you can find some way to pay to actually solve your "theoretically solved" problems. This case sounds so familiar for us researchers and all we can say is good luck with that. Or, you can overcome each hurdle as it comes up. One way or another, actually demonstrating that the hurdle has been overcome must be a priority, if the goal is to achieve some product level success. To justify these endeavors, it may be useful to study and then explain whether overcoming the hurdle may require a fundamental change in the technology, making solving it a research problem and not an engineering detail. Alternatively, the hurdles can always be included in the assumption set ("we assume/observe that the performance and cost are insignificant") as we researchers conveniently do so many times. But this option almost guarantees that the technology will never really graduate and reach the next level of success.

6. PRODUCT DEPLOYMENT

In order to obtain a thorough understanding of how the Cryptonite NXT product can work seamlessly in and protect an enterprise network, we have deployed the Cryptonite NXT security product across production networks. This section briefly describes the deployment approach as well as lessons learned with one particular production network.

6.1 General Deployment Approach

First and foremost, a strong working relationship with the IT department is of paramount importance, since any new deployment has the risk of paralyzing the operational production network. Second, the following general principles should be established and communicated with the IT staff to obtain necessary understanding before the deployment:

- minimal or preferably no disruption to users and services;
- minimal or preferably no changes to the protected operating systems (Windows, Linux, Mac, printers, other devices);
- seamless transition for users;
- incremental deployment with a progression path from small-scale and low-risk systems to full-scale and critical systems;
- instituted procedures to validate network operations at every increment;
- detailed documentation on network configuration changes, including addressing scheme, new network topology, network devices configuration, etc.;
- detailed documentation throughout the process for future improvement and deployment references.

In general, our approach to deployment consists of four phases: analysis, design, deployment, and support and maintenance.

It is necessary to analyze the network components and topology, such as the VLANs and subnets, to determine the IT department's preferred methodology for deploying any security technology within the network. This analysis can help deployment planning and not to accidentally limit access to a required business service.

Based on the analysis, it is critical to document all planned changes in the network and to obtain the final approval of the plan from the IT team. Since deployment should be executed in incremental stages, the deployment team needs to work with the IT staff to identify what is to be included in each stage of deployment. Normally, the early stages of deployment affect low-risk portions of the network. Test and evaluation plans should also be created in the design phase.

Typically, a production network should be migrated to any new technology in an incremental manner to minimize risk. The first stage is site preparation, during which any necessary configuration changes in the target site are made to the network infrastructure such as switches, routers, DNS, and directory servers and so on. The scale of these changes vary with the technology being tested.

Next, a few test endpoints are migrated to allow initial deployment acceptance testing to occur. The network should be operated in this state for a period of time to carefully analyze network behavior and performance, as well as to reduce the risk associated with further deployment. The deployment then continues following the original or now modified plan.

Once deployed, the system enters the maintenance phase. This phase is often overlooked during research, but in a deployed system this is where all of the ongoing effort and human cost associated with the technology occurs. Any daily issues that come up need efficient procedures to immediately solve them. Procedures must be in place for providing updates, resetting systems, handling failures, and a variety of other unexpected circumstances.

6.2 Penetration Test on Deployed Network

In this section we describe two separate penetration test events. First, during the process of our early deployment at a corporate production network, we performed internal penetration testing by an experienced security researcher and practitioner. The penetration testing considered three specific categories of attack: discovery, denial of service (DoS), and infection propagation. In the discovery penetration testing, standard enumeration and mapping tools for both IPv4 and IPv6 were used. Cryptonite NXT completely stopped the discovery process. No actionable intelligence about the topology or contents of the network were discovered. Additionally, MAC spoofing was attempted, but due to the network dynamics of Cryptonite NXT, the MAC spoofing attacks simply failed. In the denial of service phase of the penetration testing, the goal was to overwhelm the Cryptonite NXT network and gain intelligence about the enclave. Since the DoS packets were not correctly formed into Cryptonite NXT packets they never made it beyond the first hop, which completely mitigated the attack. The final phase of the penetration testing was to assume one endpoint was infected and monitor the spread of the infection. For this testing, modern variants of Blaster and Konquer were used. The endpoint was directly infected and the propagation command was sent to the malware. Neither malware could spread due to Cryptonite NXT's segmented attack surfaces and the malware's inability to form appropriate Cryptonite NXT packets. This testing was thorough, but an independent test was desired to provide an unbiased viewpoint and strengthen our claims.

Beyond validating functional and security aspects during an early deployment in a production network, a larger deployed network provides the opportunity for more realistic validation of security claims from an independent party's point of view. One of the top professional penetration testing companies was used for this purpose. While internal penetration testing continues to be periodically performed, an external evaluation offers an independent and fresh perspective on the technology. This fresh view also allows for a much more meaningful understanding of what current attack tactics are used in the wild, not because of the penetration tester's additional attack knowledge but because of their actions and explanations were not jaded by our internal views and preconceived notions of how we "know" an attack against Cryptonite NXT will work. We also gained an understanding of how an attacker thinks about the network and how a given security benefit actually impacts the attacker. Equally important, this external penetration testing provides an opportunity for a clean slate assessment of how the security protections could be defeated.

The professional penetration testing considered several potential actions including whether the Cryptonite NXT appliance itself could be compromised, whether an unprotected malicious device could attack the network, and whether a compromised protected device could use the technology against itself to attack the network. The experienced penetration tester used standard tactics and tricks that are effective in essentially all networks they ever tried to attack in hundreds of other enterprise networks. Additionally, once these standard attack techniques were found to be unsuccessful, a lengthy discussion ensued covering exactly how Cryptonite NXT works, what the limitations might be, and existing knowledge of possible ways for an attacker to obtain additional leverage. Additional expertise was also obtained from a larger pool of the tester's colleagues. In the end, the penetration tester concluded that meaningful discovery, scanning, and spoofing of the network were severely limited by Cryptonite NXT. If an endpoint inside the network was compromised in some way, spreading was similarly contained. Essentially, all Cryptonite NXT security claims were confirmed.

It was clear from discussions with the penetration tester that, similar to most security products, the simple knowledge of what is and is not possible can go a long way in helping to plan an attack, avoid detection, etc. We are thus leaning on the side of caution, not because any significant issues were found, but because of our belief that this knowledge would be difficult for an attacker to accurately obtain and is likely to provide many additional opportunities for an attack to be discovered very early. Therefore, since Cryptonite NXT is now an actual product, the attack details are omitted here for trade-secret and operational security reasons.

A few simple examples can still be provided. Specifically, we can discuss the two defects that were found as these are not present in any deployed systems. In one case, the penetration tester found that we lacked mitigation for a certain type of attack that we knew about but for which we did not fully understand the implications or feasibility. It was also found that our default configuration failed to encourage administrators to gain some easy security wins. As the experts of our technology, we need to include ways to guide users into maximizing its potential. Both issues were minor and have since been fixed. We also encountered cases where the penetration tester explained where our protection is in fact much stronger than we initially realized. We had not realized how commonly successful a certain type of attack was, and happily discovered that our prevention of this attack renders ineffective a tool that attackers commonly build their attack plan upon.

It is recommended that internal penetration testing should be performed on some regular basis, and professional level penetration testing should be obtained, preferably using the best testers, to provide an independent validation from an external perspective. This may shed new light on some great security benefits or configuration implications of the product.

7. CONCLUSIONS

This paper describes, at a high level, the journey undertaken and lessons learned during several years of effort to develop and productize a particular moving target defense research technology, SDNA, currently known as Cryptonite NXT.

As part of the applied research and development community, one of the common practices to gain early confidence is to execute feasibility demonstration with fast prototyping. Such effort typically requires appropriate planning and scoping of the work to be done, prioritization of the features for prototyping, leveraging previous capabilities and open-source tools, and documenting risks and limitations due to the nature of rapid prototyping.

The product development phase asks for a different mindset from team members, and dedicated program management staff and some rigorous software development life cycle process will be needed to manage the progress of a typically larger team. The importance of testing cannot be overstressed, and frequent, well-informed testing should be carefully planned and carried out against different levels of requirements, incrementally and augmented with regression tests. It is critical to completely understand the implications of the intellectual property rights for all previous capabilities leveraged as well as the open-source tools and choose carefully early on from among those that do not limit choices later on.

The security analysis of the technology needs to take into consideration various scenarios of potential attacks. Establishing an internal testbed can facilitate testing as well as interim demonstrations. The evaluation metrics should cover performance, security, and usability aspects, and such metrics should be readily measurable on the established testbed. The demonstration scenario needs to be carefully designed, and preferably use real-life examples. Candid analysis of the demonstrated capability compared to other solutions can provide deep insights about the technology and its true potential.

Drastic changes from many aspects are often needed in order to turn a technology into an actual product. Whether or not the product is better in some traditional metric becomes less important. Rather, the key question is whether or not the product can provide real customer benefits without making it harder or more expensive to deploy or maintain. A market study needs to be performed in the context of enterprise environment, the current hardware configuration and software installation, deployment requirements, interoperability with other software and security tools, and the ease of deployment and maintenance by the enterprise IT staff.

Moreover, it is also necessary to change how the technology and product are explained. It is usually a good practice to explain the product by comparing it with some familiar products. The existence of peer competition, no matter how unlikely it seems especially from behind familiar, academic lenses, can in fact help make the case with customers. An over-ambitious vision can easily lose the audience; a small, but solid and usable solution that can do an extremely good job at fixing just a handful of critical problems can carry the product a long way.

Comprehensive testing is one of the key differentiators between a prototype and a product. To transform a technology into a real product, the comprehensive test suite, which should cover all levels of the product and all aspects of functionality and security, should be carefully designed and diligently executed.

One of the key lessons learned is related to performance. Users are unwilling to sacrifice performance for security. The good and old way of illustrating performance with some arbitrarily measured throughput or latency numbers, as we typically do in research papers, can no longer be convincing in the face of the requirement for productization. A security product needs to achieve wire speeds in order to become deployable and marketable in an enterprise environment. Specifically designed and manufactured hardware and software is available and may be necessary to fulfill such needs. Similar to the spirit in the industry of real estate where location is the key factor, in our opinion the top three most critical factors for a security product to really achieve its success will be "performance, performance, performance".

At present, there is a need in the mark for concrete evidence of actual deployment and real penetration results obtained from some production networks to generate serious interest. To satisfy such needs, we have performed Cryptonite NXT deployment in a corporate production network, and invited a top notch professional company to perform extensive penetration testing on top of the deployed network. Internal penetration testing should be performed in some regular basis, and external penetration testing can provide different perspectives of the security product.

Coming from an applied research background, the authors have found that converting innovative ideas into a product is much more challenging than packaging software and selling it well. Different skills (not always technical), team makeup, processes, mindset, and even switching to a previously dimly viewed way of thinking, talking or presenting may all be necessary to tackle various aspects of productization. The authors did not have prior exposure to all of these aspects, and had to struggle through the journey and learn from experts in relevant areas. It has been a rough yet extremely rewarding journey, and it is our hope that the lessons learned could help other aspiring researchers to turn their innovative ideas into great products, which will collectively enhance the overall security of computer systems and networks.

8. ACKNOWLEDGMENT

We are grateful to the opportunity offered by the organizing committee of the moving target defense workshop 2016 to share our journey and lessons learned. Our special thanks are due to Air Force Research Laboratory (AFRL), who initiated the related research program and sponsored our research and development effort, without which the reported journey could not have started.

9. REFERENCES

[1] Gordon, R., Transforming Innovative Security Concepts into Disruptive Security Companies, Invited Talk, In Technical Sessions of 24th USENIX Security Symposium, 2015

[2] Dunlop, M., Groat, S., Urbanski, W., Marchany, R., and Tront J. 2012. The Blind Man's Bluff Approach to Security Using IPv6, *IEEE Security & Privacy*, vol.10, no. 4, pp. 35-43, July-Aug. 2012, doi:10.1109/MSP.2012.28

[3] Durumeric, Z., Wustrow E., and Halderman, J. 2013. ZMap: Fast Internet-wide scanning and its security applications, In *Proceedings of the 22nd USENIX Security Symposium,* 2013.

[4] Jajodia, S., Ghosh, A.K., Swarup, V., Wang, C., Wang, X.S. (Eds.). 2011. *Moving Target Defense: Creating Asymmetric Uncertainty for Cyber Threats*, Springer, 2011.

[5] Jajodia**,** S., Ghosh, A.K., Subrahmanian, V.S., Swarup, V., Wang, C., Wang, X.S. (Eds.). 2013. *Moving Target Defense II: Application of Game Theory and Adversarial Modeling,* Springer, 2013.

[6] Okhravi, H., Rabe, M., Mayberry, T., Leonard, W., Hobson, T., Bigelow, D., and Streilein, W. 2013. Survey of Cyber Moving Targets, *Massachusetts Institute of Technology Lincoln Laboratory, Technical Report 1166*, 2013.

[7] Yackoski, J., Xie, P., Bullen H., Li, J., and Sun, K. 2011. A Self-Shielding Dynamic Network Architecture, In *Proceedings of the Military Communications Conference 2011.* 7-10 Nov. 2011. DoI 10.1109/MILCOM.2011.6127498.

[8] Yackoski, J., Bullen H., Yu, X., and Li, J. 2013. Applying Self-shielding Dynamics to the Network Architecture, in *Moving Target Defense II: Application of Game Theory and Adversarial Modeling*, S. Jajodia Ed. Springer, 2013.

[9] Yackoski, J., Li, J., DeLoach, S., Ou, X. 2012. Mission-oriented Moving Target Defense based on Cryptographically Strong Network Dynamics, in *Proceedings of CSIIRW*, 2012.

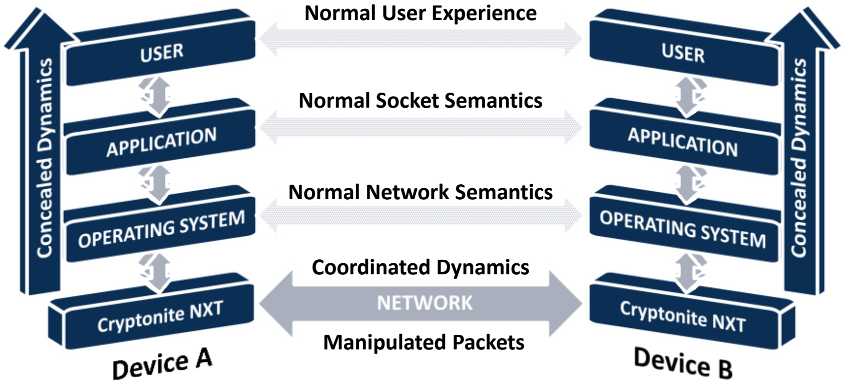

Figure 1: Semantic view of SDNA/Cryptonite NXT.

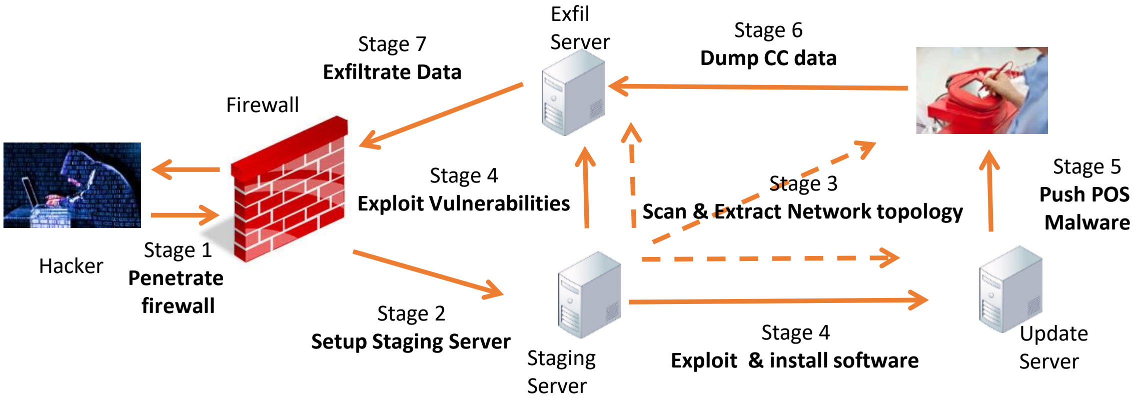

Figure 2: Attack stages in Target scenario.

Table 1: Attack summary for Target scenario.

	Attacker's Action	Expected Impact	Target's Network	FireEye Protected	Cryptonite NXT Protected
1	Attach laptop with malware and stolen credentials	Hacker is able to penetrate the firewall	**Success**	**Success (Physical Access)**	**Success (Physical Access)**
2	Use credentials to get staging server	Use this server as the point of entry into the network	**Success**	**Success**	**Attack Fails**
3	Scan the network to extract topology	Locate servers of interest for the rest of the attach and export	**Success**	**Success** **Warning "unknown binary"**	**Attack Fails**
4	Install Software on Servers	Upload malware to update server. Set up a dump server	**Success**	**Success** **Warning "unknown binary"**	**Attack Fails**
5	POS Terminal software update	Malware is pushed to the POS terminals, collect credit card #s	**Success**	**Success (Legitimate Operation)**	**Success (Legitimate Operation)**
6	Transfer data to dump server	Collect credit card #s from POS terminals to prepare for export	**Success**	**Success**	**Attack Fails**
7	Export data	Software setup during exploit allows data to be transferred	**Success**	**Success**	**Attack Fails**