Effectively Testing APT Defences

Defining threats, addressing objections to testing and suggesting some practical approaches

Simon P.G. Edwards
Dennis Technology Labs
simon_edwards@dennis.co.uk

Richard Ford
Florida Institute of Technology
rford@se.fit.edu

Gabor Szappanos
Sophos Ltd
gabor.szappanos@sophos.com

***Abstract*— Anyone watching the cybersecurity marketplace will have noticed a rapid rise in products that claim to provide protection from "Advanced Persistent Threats" (APTs). As targeted attacks gain more attention, and protection developers pay more attention to the implementation of new defensive technologies, the need arises for the testing of product efficacy with respect to this new kind of threat. However, compared to general product testing, APTs present additional challenges for the testers. In this presentation, we ask if APT protection can be tested, and if so, can it be done practically?**

I. INTRODUCTION

In this paper, we address some of the challenges related to the testing of APT protection software suites and devices. Our arguments are essentially three-fold. We first look at the subjective and confused range of definitions of what an APT even comprises.

We then look at some of the objections raised to testers' measurements of APT protection efficacy in light of these definitions. Finally, we offer some simple guidelines for those who are attempting to construct or interpret tests of APT protection.

Our conclusion is that, while the entire APT space suffers at the hands of definitional uncertainty, there is a workable way forward for tests that measure different aspects of APT protection. With sufficient effort tests can measure end-to-end protection.

Such tests are ambitious but, given the importance and cost of both potential APT breaches and APT defences, a more scientific approach must be taken.

II. WHAT IS AN APT?

Although the term 'APT' is used commonly nowadays, there is no generally accepted definition for it, and this contributes greatly to the problem of testing.

In part the APT has become this year's buzzword but vendors, reviewers and users employ the term differently depending on circumstance and goal. Such definitional challenges only add to the confusion.

For example, TechTarget uses the following definition:

"An advanced persistent threat (APT) is a network attack in which an unauthorized person gains access to a network and stays there undetected for a long period of time." [1]

According to this definition the sample must be undetected to be an APT. If a product detects a threat, then it is not an APT.

This leads us to the inevitable outcome that the only valid outcome of a test of APT protection is that nothing is detected (otherwise the test sample is not an APT).

While this definition therefore significantly simplifies APT testing in general, it would make APT testing a very simple (non-existent) task so we should aim for a more practical one. Here are a few more definitions that are quite interesting:

1) Wikipedia

"APT is a set of stealthy and continuous computer hacking processes, often orchestrated by human(s) targeting a specific entity. APT usually targets organizations and or nations for business or political motives. APT processes require high degree of covertness over a long period of time." [2]

2) NSS Labs

NSS Labs adopts an alternative acronym for a targeted attack, referring to a Targeted Persistent Attack (TPA).

"Targeted: The attacker selected the organization, for a specific reason.

Persistent: The attack is capable of using multiple command-and-control channels and attack vectors, and constantly increasing its penetration of your IT systems and resources. It is also stubborn, resisting remediation attempts.

Attack: While the word 'threat' is somewhat nebulous when used in the context of APT, there is nothing unclear about it here. This is a true attack, and it may have several distinct stages." [3]

3) Gartner

"Advanced threat - any attack that gets past your existing defences.

Persistent threat - any successful attack that goes undetected and continues to cause damage.

Advanced persistent threat - any attack that gets past your existing defenses, goes undetected and continues to cause damage." [4]

The problem with these definitions is, once again, that they attribute being undetected to being a core feature of an APT. This definition renders APT defences and tests useless. Other definitions focus on other aspects:

4) RSA

"An Advanced Persistent Threat (APT) is a targeted attack against a high-value asset or a physical system." [5]

While this is a useful definition that makes it easy to determine if an attack belongs to this category it does not explain the significance of the "Advanced" and "Persistent" attributes of an APT.

5) Damballa

"Advanced Persistent Threats (APTs) are a cybercrime category directed at business and political targets. APTs require a high degree of stealthiness (sic) over a prolonged duration of operation in order to be successful…

Advanced – Criminal operators behind the threat utilize the full spectrum of computer intrusion technologies and techniques…

Persistent – Criminal operators give priority to a specific task, rather than opportunistically seeking immediate financial gain…

Threat – means that there is a level of coordinated human involvement in the attack, rather than a mindless and automated piece of code…" [6]

We have a lot of definitions that attempt to define APT on an abstract level, hardly helping testers to categorize test scenarios. Our best option at this point is to change scope and deal with a better defined and practical definition of targeted attacks along the lines of the RSA definition.

The terms 'APT' and 'targeted attack' are often used synonymously by the press and the APT protection providers so it makes sense to stick to the easily definable 'targeted attack' cases in test scenarios.

For practical purposes of testing we will define targeted attacks as:

A targeted attack is an infection scenario executed against a limited and pre-selected set of high-value assets or physical systems with the explicit purpose of data exfiltration or damage.

III. WHAT DOES AN APT ATTACK LOOK LIKE?

Given that we know now what an APT is for all practical purposes, we have still to explore what such a targeted attack might look like.

According to Mandiant [7], an APT attack is more of a campaign than a single event, which follows the following rough outline:

- Reconnaissance — prior to performing the attack, information is gathered that is used as part of the social engineering repertoire during the later stages.
- Initial compromise — performed by use of social engineering and spear phishing, over email and/or by planting malware on a website that the victim employees are likely to visit.
- Establish Foothold — plant remote administration software in victim's network to create network backdoors and tunnels allowing stealth access to its infrastructure.
- Escalate Privileges — use exploits and password cracking to acquire administrator privileges over victim's computer and possibly expand it to Windows domain administrator accounts.
- Internal Reconnaissance — collect information on surrounding infrastructure, trust relationships and Windows domain structure.
- Move Laterally — expand control to other workstations, servers and infrastructure elements and perform data harvesting on them.
- Maintain Presence — ensure continued control over access channels and credentials acquired in previous steps.
- Complete Mission — exfiltration of stolen data from victim's network.

A. Practical example of a targeted attack scenario

For the illustration of a targeted attack workflow, we picked up an attack that exploited the popular Japanese word processor Ichitaro.

This attack scenario happens in many consecutive steps but will not involve some of the steps in the previous list (initial reconnaissance, lateral movement). Regardless, it gives a good technical overview of what happens on an attacked system in the background.

a) Phishing e-mail

The malware is delivered in e-mail. From the header, the email appears to have been sent from AOL webmail.

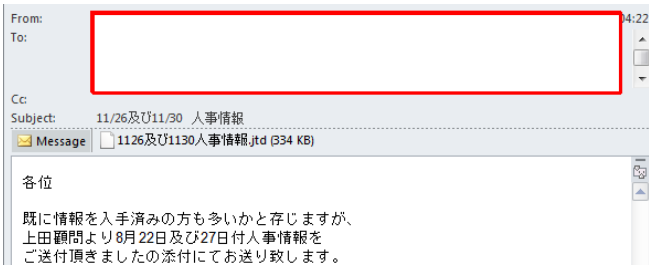

Fig. 1. The phishing email that allows the attacker entry to the system.

The following is a translated version of the message body (see Fig. 1.): “Dear Although you propose those already obtained also whether there is many information already, 27 date personnel information and August 22 from Ueda adviser. We will send you an attachment of had you sent. Much for your confirmation, thank you.”

b) Exploited document

The attachment is an RTF document that exploits a vulnerability in Ichitaro, a popular word processor application in Japan. Upon opening the document the vulnerability is triggered and the ‘shellcode’ that is embedded in the document is executed.

c) Shellcode

The shellcode is encoded with the standard unicode_upper encoder of Metasploit framework.

The shellcode extracts and executes the embedded executable. It is encrypted with a simple one-byte XOR algorithm but, to make the encryption key less obvious, the 0 bytes and the key bytes are left intact. If it was not done this way then the large blocks of zero bytes that are normally present in PE files would convert to the value of the key byte and simply looking at the file for a large block of similar bytes would reveal the encryption key.

d) WinRAR SFX installer

The shellcode extracted, decrypted and executed a WinRAR self-extracting archive.

The archive contains three files (see Fig. 2.), of which starter.exe (a clean digitally-signed executable) is set to execute automatically (and silently) after unpacking. The malware loader that is loaded by the DLL search order vulnerability is called splash_screen.dll. It is popular in the Plugx malware family tree. The splash_screen.dll.sp1 file is the encrypted payload file loaded by splash_screen.dll.

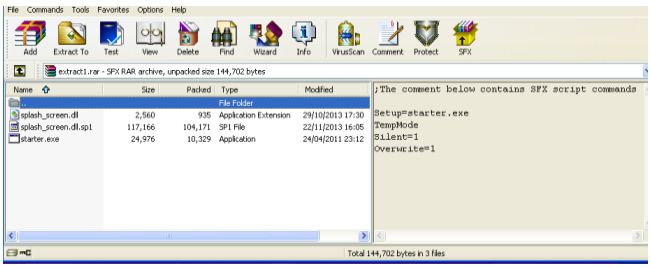

Fig. 2. The contents of the packed archive used in the attack.

e) splash_screen.dll

This is a simple loader. On execution it checks the date and, if it is before 08/08/2013, aborts. Otherwise it reads the payload file from the hard drive into the memory and transfers execution to it, starting from the first byte.

f) Payload loading

The beginning of the payload code decompresses the final backdoor using the LZNT decompression algorithm.

The decompressed content is nearly a DLL file. The magic bytes, marking the start of the exe header ('MZ') and the start of the Portable Executable header ('PE') are both overwritten with 'XV' (see Fig. 3.). It makes the unpacked image non-executable for the operating system (and very hard to analyze properly).

The lack of the markers does not bother the malware loader as it does the PE loading by itself and ignores this fact:

- It allocates the memory areas for the PE section, sets the access rights and copies the section content there
- Next it performs the relocations needed for dynamically linked libraries
- It then resolves the required API functions from the import table

It overwrites the beginning of the file (including the PE header) with zeros. This way the dumped executable will be very difficult to analyse (normally this step in older Plugx variants was done by the final payload itself, after loading). Finally it jumps to the entry point of the payload file and executes it.

g) Final payload

The final payload is a backdoor that connects to a remote server.

Functionalities include: Collect running process and module information; Manipulate/create services; Create/delete process; Create/delete files; Create/delete Registry entries; Gather system information (memory, system time, LCID, system locale); Get disk/volume information, free space; Keylogging; List network resources

Fig. 3. Note the beginning of the file is missing the PE designation, in an attempt to evade detection.

and connections; Shutdown/reboot/lock workstation; Create screenshot.

Attackers can use the services provided by the Plugx backdoor to start the final stage of the infection scenario, that being data exfiltration.

h) Layered defences against targeted attacks

As can be seen, the attack described above has multiple different steps. A modern anti-malware solution typically features several modules that utilize different detection approaches to provide a multi-layered diverse solution. This makes it more difficult for malware to bypass defences.

These modules could include the following:

- Application Control: block the execution of potentially unwanted/unauthorized applications
- Anti-Spam: block bulk e-mail
- Scanner: specific detection for known malware; generic detection for new malware
- Firewall: blocks outbound communication attempts and inbound attacks
- IPS: packet level filtering of network traffic
- URL filtering: reputation or blacklist based URL blocking
- DLP: prevents exfiltration of sensitive data
- Exploit protection: detect exploitation of application vulnerabilities
- Behavioural-based detection: detect malware based on runtime activities in the system

These different methods provide defence in depth against targeted attack scenarios, giving opportunities to detect and block the scenario at multiple points. In the example detailed in the previous section the different modules can block the scenario at several points, as illustrated below in Fig. 4.

Any test that uses only a subset of the above detection capabilities is an incomplete test and, as such, does not give the full picture on the protection capabilities of the particular solutions being assessed.

As an example, in a test in which only the VirusTotal detection result of the final payload binary is used, several of the multi-layered defence modules (showed as dimmed in Fig. 5.) are ignored:

Thus, testing end-to-end protection is critical if a broader view of product efficacy is to be gained.

IV. Objections to APT testing

There are a number of reasons why vendors of anti-APT products may object to testers evaluating them. There are public statements criticizing tests in the press and earlier in this paper we explored some definitions of APTs that clearly show vendors, analysts and others having very different opinions on what an APT actually is.

If testers and vendors can't agree on what products are supposed to do then this brings a major obstruction to testing. Here are some reasons why vendors may object to testers evaluating their products:

1) Historic skepticism about security testing/testers in general.

2) The APT detection/protection market is young and very sensitive to test results.

3) Tests are too limited to provide useful results because they are not based on what is happening in the 'real world'.

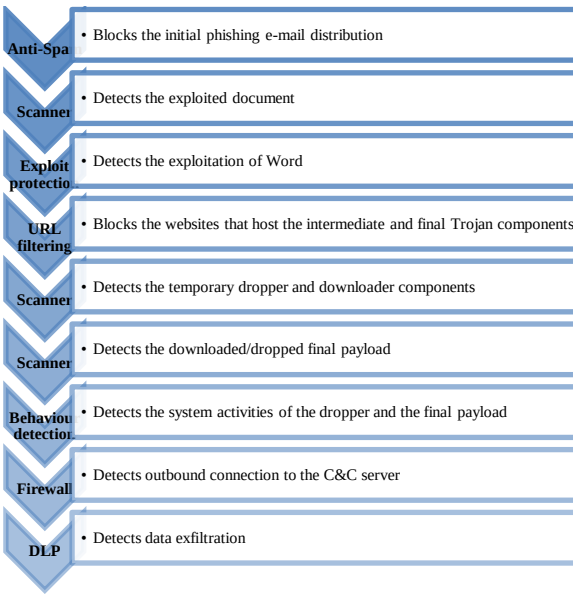

Fig. 4. Different security modules can block the scenario at several points.

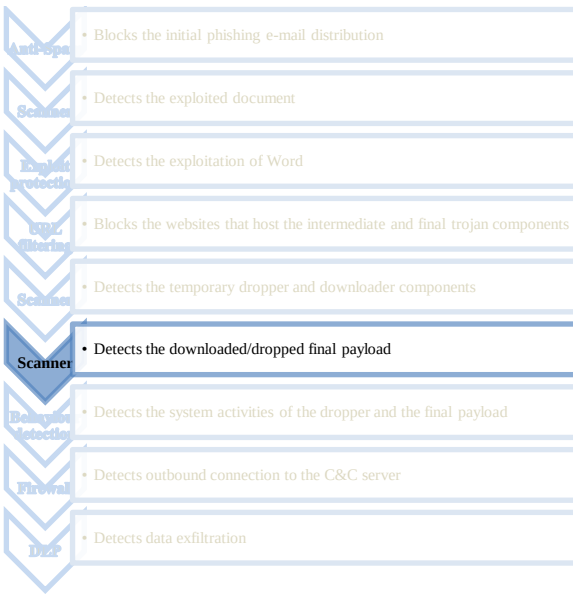

Fig. 5. Ignore multiple defense layers and risk an incomplete test.

4) Claims that products are not testable because a realistic test requires:

a) defenders to react to alerts.

b) unknown malware/exploits.

c) malware/exploits that will always bypass other defensive technologies.

B. Commentary

1) Security testing/testers are dishonest and/or incompetent

"Testers know nothing about advanced threats, how enterprises manage their networks and are too focused on traditional anti-virus testing methods to be able to test anti-APT products properly."

This theoretical objection to anti-APT product testing may be the result of a reflexive response to standard methods of testing anti-virus software and the reputation of well-known anti-virus testers.

Some who today work in anti-APT businesses used to work in the anti-virus industry, which is a notoriously small world. It is possible that these people may be prone to assuming (based on historic experience) that anti-virus products do nothing more than scanning files to generate signatures, which are then compared to white/blacklists.

This definition of anti-virus was probably in the mind of the Symantec employee when he told The Wall Street Journal in May 2014 that "anti-virus is dead" .

From a business perspective it also suits anti-APT businesses to define anti-virus as being mainly or wholly signature-based because they can then demonstrate that their more sophisticated product(s) provide better protection.

Ex-anti-virus people probably remember the dissatisfaction there is/was with anti-virus tests that simply scan files rather than executing them or, even better, downloading them from live sources in real time. It took the anti-malware testers a long time to update their methodologies and this reticence damaged testers' reputations in general and not just the reputations of the testers who refused to change in good time. This damage may persist in the minds of some. There may also be lasting memories of conspiracy theories in which testers rig results for commercial gain.

If vendors assume that anti-virus means scanning files and comparing signatures, and that testers limit themselves to using on-demand scanning solely as a methodology, then it's not surprising that they would object to such traditional and limited assessments of their sophisticated products. If the vendors additionally believe that some testers are actively dishonest in representing the results then the objections seem even more reasonable.

However, today anti-virus products include a much wider set of protection features and many testers have adopted relevant testing methodologies to take these into account. Some also take very open approaches to disclosing methodologies and test data.

Of course, it may not suit anti-APT vendors to acknowledge this progress in the testing world because if they do then tests gain credibility and, without credible test results, vendors are able to make uncontested claims about their products.

2) The anti-APT market is quite sensitive to test results

The advanced threat detection industry is relatively new, in terms of how it markets itself. In the follow example we examine FireEye's positioning and technology, but the same argument could be made for any number of new vendors in this space.

According to marketing material, FireEye offers what sounds like a novel proposition to detect or prevent advanced attacks and has been apparently successful in marketing its approach.

Whether or not FireEye's combination of sandboxes and signature-based anti-virus scanners is very different to offerings from other security vendors doesn't much matter currently because it is perceived by the market to be different.

FireEye makes a clear effort to differentiate itself from more 'traditional' companies and includes quotes on its website such as the following (with our *emphasis*):

"FireEye is clearly the *next generation* of network security. Their *unique signature-less technology* is simply unmatched in detecting and *blocking* this new breed of advanced malware. (Director of security - global financial services company)"

The emphasized key phrases clearly show an effort to distinguish FireEye's approach as being different to traditional anti-virus technology. The statements claim a lack of reliance on file signatures (even though its product actually includes an anti-virus scanner) and imply that there is a new type of malware that will always evade other solutions, in total contrast to FireEye's solution.

At this early stage in the game positive test results would help these new companies build market share and stock price, while negative results are almost certain to damage the companies fast and hard. These "next generation" products are largely unproven and so are vulnerable to doubt from potential customers and investors. As such it may be better, from the vendors' point of view, for there to be no tests running at all. As proposed above, a lack of results allows for uncontested marketing claims.

3) Tests are not 'real world'

Anti-APT products are complex and involve monitoring real network traffic, servers and other endpoints for attacks. As such one might expect a realistic test to completely reproduce similar networks and attacks. This is unlikely to be feasible for even the most well-resourced testing organization.

That being the case, FireEye has concluded that testing is impossible and that the best solution is simply to buy its solution. FireEye CTO Dave Merkel told CRN that, "the

best way to evaluate FireEye is for organizations to deploy our technology in their own environment and they will understand why we are the market leader in stopping advanced attacks."

But what is the 'real world' when it comes to threats and protection against those threats? Is it possible to at least test some very important parts of anti-APT products? Or should testers give up and leave customers to simply trust vendor claims?

To continue with FireEye's statement, Merkel mentioned "sophisticated criminal networks and nation states" as being the perpetrators of APT attacks. This leads us to two important questions:

1. Is an APT attack the exclusive domain of ultra-sophisticated and extremely well-resourced actors?
2. Is the attack only an 'APT' if those actors always use the most technically advanced attacks? Does it lose it's 'APT' label if the same actors use less sophisticated tools, tactics and techniques?

If the answer to both of the above questions is "yes" then a government intelligence agency could target competitors using relatively unsophisticated techniques and, in doing so, would evade the 'APT' classification and, presumably, the anti-APT products on offer.

A "yes" answer would also lead us to conclude that opponents of an organization that are neither part of a criminal organization nor a government-sponsored group could use extremely advanced tools and tactics to steal information but would be able to evade the 'APT' classification and, again presumably, the anti-APT products/services.

Of course, if APTs are the exclusive domain of spies and the Mafia then, regardless of how sophisticated they are, testers will never be able to launch an APT and the testing game is over. That is unless a tester becomes a spy or master criminal…

Let's have a sanity check:

a. A national intelligence agency sets up a watering hole attack using a commercial exploit toolkit in order to gain long-term access to political activists. It uses the toolkit to appear less sophisticated for disavowal purposes when the attack is inevitably uncovered in time.

b. An individual discovers a software vulnerability and develops a zero day exploit, which s/he uses to gain and maintain access to a network. This access is used to create deeper levels of access to systems on the network, which is then used to steal data on a regular basis. The data may be of interest to that individual for a number of reasons, possibly to sell or to use directly.

Are either of the above examples not APTs and, if not, why not? Customers of anti-APT products would most likely not care about subtle labelling. They would expect both of the above events to be detected and ultimately curtailed, if not prevented in the first place.

4) Tests require defender reactions

This is a genuinely tricky element of any enterprise-level technology test. In a real working situation anti-APT products do not automatically mitigate all threats. There most likely has to be human intervention from the defender's side, such as an administrator noticing and acting upon one or more alerts.

How this person or persons behave will vary depending on their resources such as the level of their training, their technical skills, commitment and the amount of time they have available to analyze and react to alerts.

Testers will need to replicate the behavior of one or more operators of an anti-APT solution realistically. For useful results it may require multiple tests with different operator behaviors. This could turn into a behavioral test more akin to a social studies project instead of the technical test most would expect.

5) Tests require unknown malware/exploits

If the right type of malware and exploits for an anti-APT test have to be unknown then how is it possible for a tester to conduct such a test? Possible methods might include creating new threats or discovering threats that are live on the internet that are not detected by any known anti-malware product.

Creating new malware requires a set of skills not usually found in anti-malware testing labs, although such tests have been known. For example, Simon PG Edwards has performed consumer computer magazine tests in the past that used heavily modified Trojans , while MRG Effitas regularly conducts tests using 'simulators' (modified Trojans) and even live botnets .

However, in all of these cases the threats are at least based on known malware and it is unlikely that there are many testers competent enough to create complete new and sufficiently sophisticated malware. It may not be necessary to do so, though.

Tests should be able to assess claims made about the anti-APT products, which are supposed to be able to handle advanced attacks and malware. How advanced these elements should be will depend on how advanced they are in real life and recent reports suggest that APTs in the real world often compromise a chain of events and threats that are quite easily copied by testers.

For example, in September 2014 FireEye published a blog post detailing an APT campaign by the so-called APT12 "cyber espionage" group [8]. The tools, tactics and techniques used should be familiar to most in the anti-malware testing industry and are not especially sophisticated in terms of unknown malware and exploits.

In fact the campaign in question used an exploit toolkit to attack a two year-old vulnerability (CVE-2012-0158). The tactics used involved attaching an 'infected' Word document to a spear-phishing email.

It does not seem that this particular campaign was so advanced that something very similar could not be run in a test.

6) Tests require malware/exploits capable of bypassing other solutions

This objection, which stems from FireEye's main marketing message at the time of writing , is quite clever because it is sort of meaningless but also essentially excludes any malware that can be detected by any other anti-malware solution. This in turn raises the bar for testers to the point where they have to find completely new malware that has never been encountered by any known security vendor.

The criteria is also quite meaningless, though, because it's not so much the case that malware can bypass protection mechanisms but more that protection mechanisms may fail to detect or protect against the malware. This sounds like nonsensical semantics so let's explore what this subtlety really means.

Malware is not usually capable of 'bypassing' an anti-virus product. A dropped file would not, before it executes, usually be capable of very much at all. Anti-virus products can, however, miss malware.

Let's imagine a malware binary that is not known to any anti-malware vendor. This would fit the above criteria for inclusion in a test if it could be proven to be universally undetectable. (Let us, for the minute, forget how hard it would be to confirm for sure that no anti-malware product in the world could detect it without alerting anti-malware vendors as to the sample's existence.)

In another example let's consider a malicious program that is detectable by many popular anti-malware products but not by Microsoft System Center Endpoint Protection. Does this sample fit the testing criteria? After all, it can bypass at least one solution so maybe it is valid to use this as part of a test attack. This is where things become meaningless. Does the malware need to be undetectable by one, four or all other solutions before it becomes a valid candidate for a test? How unknown does it need to be, to be unknown?

In the example of the second sample, maybe it is unknown only to Microsoft today but what about tomorrow? Does it become ineligible for inclusion in the test when vendor sample-sharing systems catch up and the data reaches Microsoft?

A vendor might complain that testers should not assess its product using regular malware because that's not what it's supposed to protect against. But it's fair to test a car's safety features by driving it into a wall because, although you're not supposed to, people want to know what happens when such events happen. It doesn't mean that the product is useless if it misses some regular malware that should be picked up by another solution, and it doesn't mean that the car is useless if it crumples under the impact of the collision.

7) A thought on customer expectations

If customers expect a SourceFire or Palo Alto product to protect the network all the time by blocking new threats they may be disappointed to learn that these products do not offer this as their main service, although this is not to say they cannot still effectively defend against an APT.

Most such products monitor the network and, after detecting a threat that initially evaded them, can correlate data to discover what happened in the past.

The term 'block' is used a lot in marketing material and implies initial blocking, whereas in fact it may refer to blocking the threat after it has existed on the network for some time. This is an important factor when formulating a suitable testing methodology.

8) Obstacles vs. practical and ambitious solutions

Table 1 includes a list of some of the obstacles that we (informally) hear for APT defense testing. We offer two options for overcoming these roadblocks. The first takes a practical approach that addresses the roadblock while the second is a more encompassing approach that illustrates the 'high end' solution that is probably financially impractical but which would represent a 'gold standard' in APT testing.

TABLE I. APT DEFENSE TESTING OBSTACLES

Obstacle	Solution #1 (practical)	Solution #2 (ambitious)
All tests will not be real-world.	Use same tools, tactics and techniques as adversaries use in real-world APT campaigns.	Testers administer multiple defense systems on real target networks.
Attackers will eventually gain access if persistent so testing becomes a penetration test that necessarily requires the reaction of the target's IT staff.	Assess capabilities of automated elements and assume/categorize levels of skill, commitment and resources of the target. There may be more than one set of results for each product/attack. Assess usability and thoroughness of alert and logging system.	Perform a 'Capture the Flag' contest with independent penetration testers as the attackers and vendor-supplied consultants as the defenders (who are under pressure to perform some duties unrelated to monitoring logs, perhaps).
Testers require a range of attack skills beyond sourcing and executing malware from various sources.	Training on a range of attack skills or create bespoke environments with known weaknesses.	Sub-contract attacks to professional penetration testers.

Fig. 6. Roadblocks to good testing of APTs and potential ways forward.

V. THE WAY FORWARD

Based on our discussion it is clear that there is a continuum of possible attacks all of which fit someone's definition of what an APT is. Thus tests will, by necessity, reflect this spectrum where, as we put it, the attacker's skills range "from zero to Neo".

A. Be clear on the APT actually being tested

Based on this discussion, our first requirement for any APT test is that the definition of APT is clearly stated and

the threat model described. For any test of APT protection this step is a must.

Beyond this, it is important that the tester details what is to be tested. Is the tester claiming that the test covers the entire lifecycle of the APT, or does it handle just one or two steps? If the former (and we doubt that many testers are equipped to conduct such an all-encompassing test) then scoring becomes a huge issue. Prima facie, the earlier in the lifecycle that the threat is detected, the better. Thus, for tests that attempt to cover the range of detection options offered by APT protection test suites (because any real APT remediation is a multi-step process) look carefully at how the tester has scored detection at different layers.

B. Be clear on whether the threat is zero, Neo or other

We still see attacks using two-year old exploits and we have threats such as Stuxnet that are loaded up with zero-day attacks. When looking at a test of APT solutions make sure that the tester is clear on what level of protection is offered and from what attack. Is the test against custom malware or is it simply a Metasploit-generated year-old exploit?

While the idea of testing with zero-day attacks sounds impossible in fact there is a way in which such a test can be done. Instead of finding zero-day vulnerabilities in existing code, insert new vulnerabilities into open source software and then build an exploit for this weakness. This approach has the advantages that it creates no new threat to the general public (the exploit only works for our custom applications) and it truly tests a product's ability to protect from completely unknown threats .

C. Be clear on whether this is a test of a layer or a suite

One of the mistakes it is easy to make when evaluating APT protection is to reduce the protection range and focus on just a single aspect of protection, such as exploit detection. While such tests are valid it is important that we do not reduce the problem to just one layer of protection. For example, consider two hypothetical products: Product A and Product B.

Product A has outstanding exploit detection, but little else. Product B has a wide range of protection techniques, none of which is perfect but, when combined these layers, provide an extremely solid prevention and detection platform. Given that APTs can come in many forms (and some do not even have to use exploits) B probably has better protection than A in general. However, for some types of customer Product B reflects the 'best' component in their home-built layered protection scheme.

The guidance here is simple: be sure you understand what rolled up scores represent and do not blindly accept rankings that may not measure what you want. Furthermore, any test that claims to be testing for APT protection must consider all layers otherwise it is not a valid test of what it claims (although it may be a valid test of specific functionality).

D. Any APT test must examine infiltration

We argue that the earlier in the attack chain that the APT is caught the better. Thus, for a layered test, it is important that testers examine the most common infiltration mechanisms. This would include, at a minimum, email, web, exploit, offline (media), social engineering, file execution and downloader/droppers, which is quite a list.

VI. CONCLUSIONS

APT testing is complicated enough without having to consider the fact that the term APT is ill-defined. Furthermore, in a real APT scenario the attacker will not blindly throw in the towel when an attack is blunted, but will probe until a weakness is found. This co-evolutionary aspect between attacker and defender is most obvious when considering targeted attacks, as the attackers have specific goals and desires and will adapt to defences as best as they are able. Stopping the first salvo in an APT attack is not the end of the road.I Instead the protective countermeasures must withstand multiple waves of attack.

We argue that the single most important step with respect to moving forward is for testers and test consumers to be clear on the purpose of the test and whether that test can measure the feature(s) in question. This one single question can help both in test creation and interpretation.

The authors note that the views expressed in this paper are their own personal opinions and do not necessarily reflect the views of either AMTSO or their current employers.

VII. REFERENCES

[1] advanced persistent threat (APT) , TechTarget, November 2010, http://searchsecurity.techtarget.com/definition/advanced-persistent-threat-APT

[2]Advanced persistent threat, Wikipedia, http://en.wikipedia.org/wiki/Advanced_persistent_threat

[3]The Targeted Persistent Attack (TPA), NSS Labs, August 19th 2012, https://www.nsslabs.com/blog/targeted-persistent-attack-tpa-when-thing-goes-bump-night-really-bogeyman

[4] Defining the "Advanced Persistent Threat", Gartner, November 11th 2010, http://blogs.gartner.com/john_pescatore/2010/11/11/defining-the-advanced-persistent-threat

[5] Sherlock Holmes and the Case of the Advanced Persistent Threat, Ari Juels and Ting-Fang Yen, RSA, 2012, https://www.usenix.org/conference/leet12/workshop-program/presentation/juels

[6]Advanced Persistent Threats: A Brief Description, Damballa, https://www.damballa.com/advanced-persistent-threats-a-brief-description

[7] Phases of an APT attack, Mandiant, http://intelreport.mandiant.com

[8]Darwin's Favorite APT Group, FireEye, https://www.fireeye.com/blog/threat-research/2014/09/darwins-favorite-apt-group-2.html