Pentest as a Service with Cobalt

Matt Bromiley

A SANS Product Review

Pentest as a Service with Cobalt

Written by **Matt Bromiley**

March 2021

Sponsored by:

Cobalt.io

One of the most popular forms of security control testing is penetration testing, or "pentesting." Often contracted out to a third party, many organizations rely on pentests to help them identify weaknesses and vulnerabilities within their security posture. This concept is sound in nature—let an experienced professional test your environment and report back faults so you can improve for the future. In theory, yes. Unfortunately, it is easier said than done.

There are old stigmas that prevent many pentests from successfully changing an organization's security posture. Whether it's an outdated approach, reduced scope, or an inefficient process, something is not working. In our personal experience, it's far too often that an organization is breached, only to discover that a pentest identified the entry vector months before.

Perhaps it's time to change some of these stigmas. In this paper, we introduce you to Cobalt, a company that is changing how you schedule, perform, interact with, and act upon penetration testing results. We had a chance to spend some time reviewing the Cobalt pentesting experience and can genuinely say that this was an information security experience unlike many others. Cobalt leverages the latest trends in on-demand communication and knowledge to offer an interactive, transparent penetration testing experience.

Our key takeaways from the Cobalt experience included:

- The ease of scheduling a pentest against various enterprise assets was simple and effective, allowing us to control scope.
- The platform provided granular insights into risks identified and the subsequent impact to our environment, prioritized in an easy-to-consume "what should we fix now" format.
- During the penetration test, coordination with the Cobalt team allowed us to evaluate security controls and posture in real time.
- A detailed, impactful report clearly outlined business impact and provided a checklist for post-test remediation.

The Cobalt experience introduces a new method to perform and consume penetration tests, and we think blue teamers and defenders would benefit from this approach. This is yet another gap that Cobalt seeks to fill: How do the results of a penetration test reach defenders, and can they use those results to make a change? By introducing an interactive component to the pentesting process, Cobalt gave us a front-row seat to their operations and allowed us to obtain and provide instant feedback.

As you work your way through this review, consider the following questions concerning your organization:

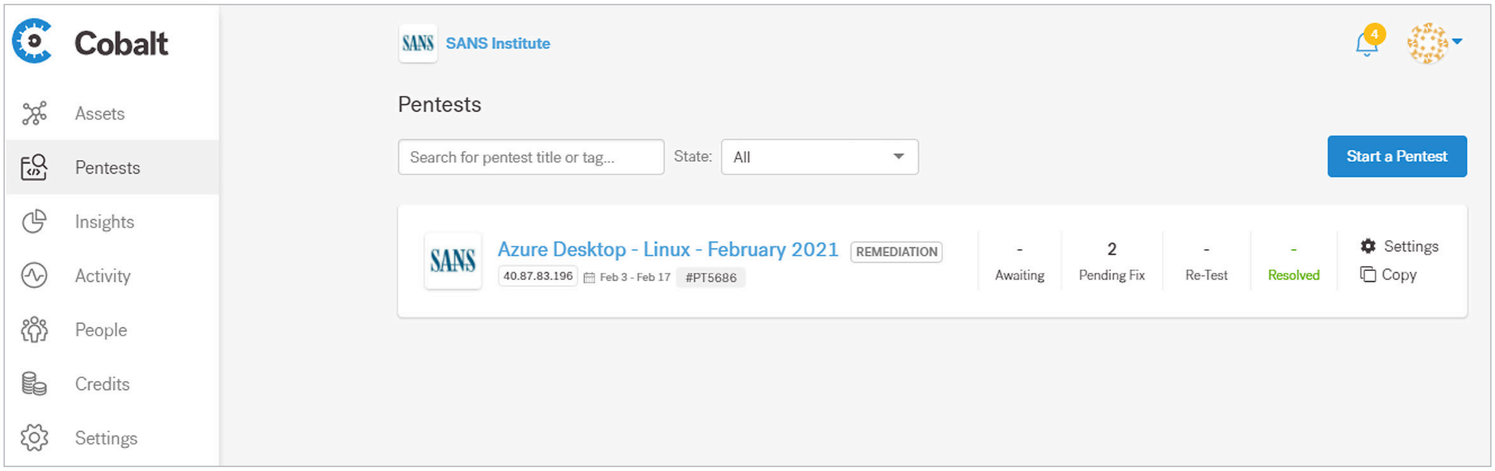
- Who is involved in the penetration testing process?
- How much time is invested in ensuring a penetration test is successful?
- How do you monitor the coverage and progress of a penetration test?
- How are the results from a penetration test observed and implemented in your organization?
- How do you ensure that pentests are consistent and, thus, repeatable to measure growth?

Security control testing is a necessary process, the value of which is determined by how you interpret and act upon the results you receive. As you'll see, Cobalt makes execution and delivery seamless, so your organization can focus on improving its security posture. Let's begin!

Getting Started with Cobalt

While the core of the Cobalt experience rests in the interactive penetration test, they also have built a functional and intuitive platform for their customers. Organizations can use the platform to define assets within the organization, schedule penetration tests, and review tests from the past, present, and future. Figure 1 shows a screenshot of Cobalt's main dashboard.

Cobalt's main dashboard is pleasantly straightforward. During our product review we conducted only one penetration test, which is clearly outlined in the dashboard shown in Figure 1. We will examine the penetration test experience in the section "The Cobalt Experience" later in this paper. However, before we begin examining results, it is important to understand how one begins their relationship with Cobalt.



Pentests

Azure Desktop - Linux - February 2021 REMEDIATION
40.87.83.196 Feb 3 - Feb 17 #PT5686

-	2	-	-
Awaiting	Pending Fix	Re-Test	Resolved

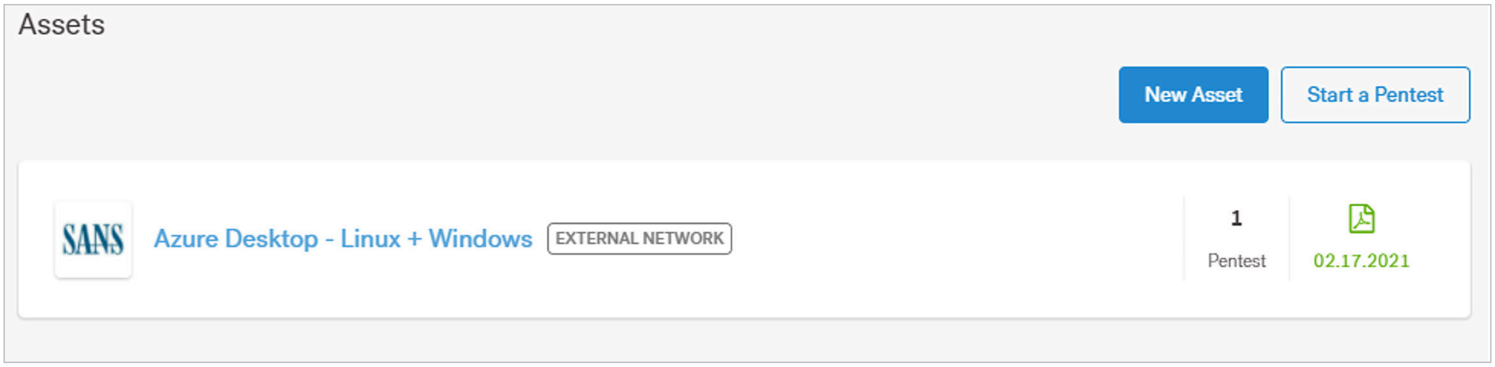
Figure 1. Main Dashboard

A typical penetration test process may involve a questionnaire or meeting(s) to transfer knowledge about the organization's environment for the purposes of the test. Traditionally, this can be a long and arduous process, especially because many organizations do not have their environments correctly inventoried. Furthermore, inventory management may not be consistent from test to test, resulting in different or incomplete scopes for each test.

> Cobalt's penetration test experience, including asset classification and scheduling, takes place primarily online. While Cobalt will happily set up a call and walk you through the steps, we found the virtual experience gave us ultimate control over scheduling and scope without any hassle.

Asset Classification Is Critical to Success

Cobalt strives to eliminate an issue of inventory first. Once we gained access to the platform, the first step was to catalog key assets within our environment. We did this via the aptly named Assets tab, shown in Figure 2.



Assets

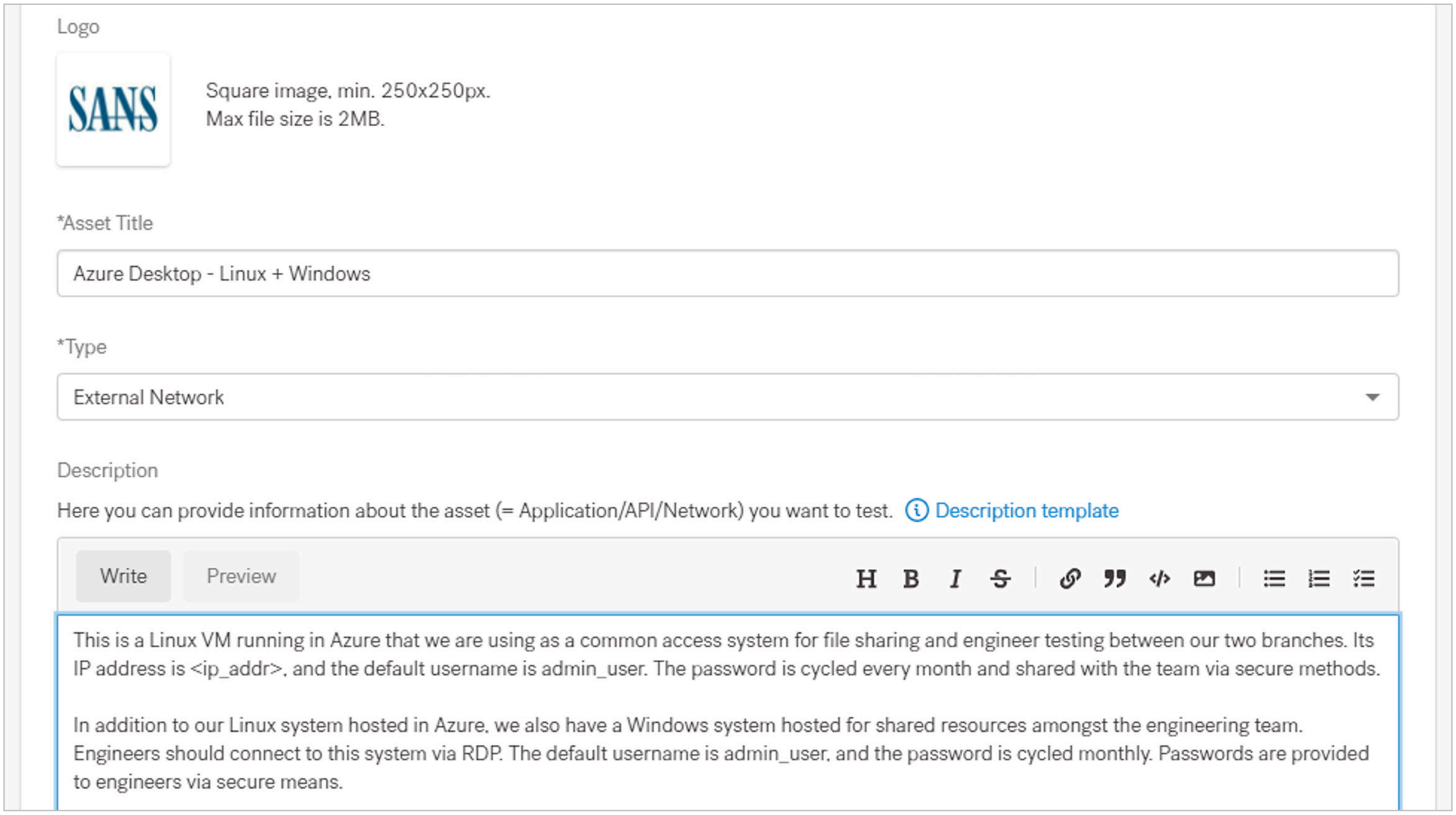
Azure Desktop - Linux + Windows EXTERNAL NETWORK

1 Pentest

02.17.2021

Figure 2. Assets Tab

Right away, we began to see value in the platform, if only from an inventory management perspective. Providing an outlet for customers to track their own inventory within Cobalt ensures that the same asset classes can be tested with consistency. We also enjoyed the freedom to describe an asset in plaintext or by dragging and dropping related documents, providing as much context as necessary for the penetration test. As depicted in Figure 3, we created a Linux virtual machine in Microsoft Azure for the purposes of this review.



Logo

Square image, min. 250x250px.
Max file size is 2MB.

*Asset Title

Azure Desktop - Linux + Windows

*Type

External Network

Description

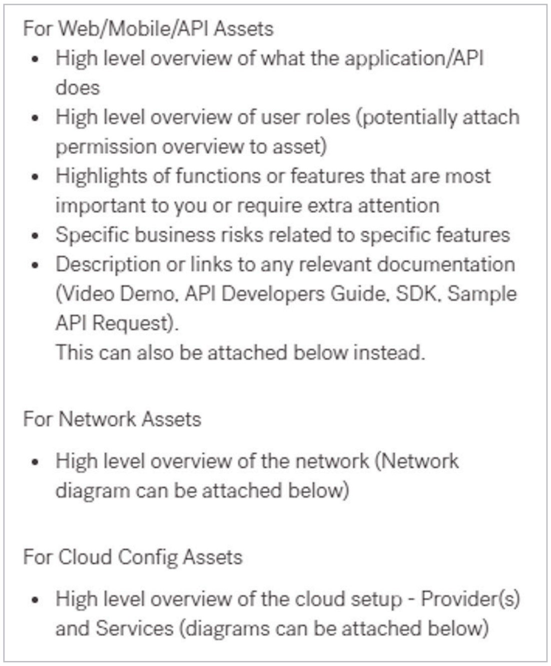
Here you can provide information about the asset (= Application/API/Network) you want to test. Description template

This is a Linux VM running in Azure that we are using as a common access system for file sharing and engineer testing between our two branches. Its IP address is <ip_addr>, and the default username is admin_user. The password is cycled every month and shared with the team via secure methods.

In addition to our Linux system hosted in Azure, we also have a Windows system hosted for shared resources amongst the engineering team. Engineers should connect to this system via RDP. The default username is admin_user, and the password is cycled monthly. Passwords are provided to engineers via secure means.

Figure 3. Asset Details for a Linux Virtual Machine in Microsoft Azure

Note that we were able to provide context on the system, what it is used for, and any details concerning credentials we want to provide to the testing team. Our example, which will appear heavily in a subsequent section, focuses on a Linux virtual machine used as a file-sharing system between two remote teams. The open Description text box allowed us to provide as much context as we saw fit. Cobalt does provide a template for descriptions, looking to garner as much information as possible so that they can structure the penetration test accordingly. Figure 4 shows a snippet of the Description template.



For Web/Mobile/API Assets

- High level overview of what the application/API does
- High level overview of user roles (potentially attach permission overview to asset)
- Highlights of functions or features that are most important to you or require extra attention
- Specific business risks related to specific features
- Description or links to any relevant documentation (Video Demo, API Developers Guide, SDK, Sample API Request).
  This can also be attached below instead.

For Network Assets

- High level overview of the network (Network diagram can be attached below)

For Cloud Config Assets

- High level overview of the cloud setup - Provider(s) and Services (diagrams can be attached below)

Figure 4. Snippet of Description Template

Figure 4 also provides details on the types of assets that can be classified within Cobalt. While we went with a simple virtual machine, the following categories can be selected:

- Web
- Mobile
- API
- External Network
- Cloud Config
- Internal Network

Cobalt also allows for combining of select asset categories for multistage or multi-asset tests. The combinations available to us included:

- Web + API
- Web + External Network
- Web + Mobile

After selecting a class and providing relevant details, an asset is created and available for testing.

You might ask, why is asset classification important for a successful penetration test? It is our belief that when an organization conducts a penetration test, it is cognizant of the environment's sensitivity. Surely, some organizations may issue an open-door policy—which we would advise against! Usually, penetration tests are scoped in a way to focus only on a specific segment, minimize potential business risk, or achieve a limited set of goals (e.g., see if you can get Domain Administrator, but do not run ransomware in our environment).

Successful penetration tests begin and end with proper asset classification. If you are asking someone to test your environment, you should have knowledge about *what* you are testing and expect to receive from the test.

Without proper knowledge of the assets they are testing, penetration testers will rely on rules and restrictions set forth by the customer to limit their tests. Who do these rules help? They provide little context to the penetration tester and, as a by-product, will limit the breadth and value of our results. What if, in the process of building an asset list, we had to describe the value to the organization and why it would or would not be in scope?

With Cobalt, we found that in having to describe an asset, we did exactly that. Basic questions that many organizations rarely ask themselves include:

- Is the asset in production, development, or testing environments?
- Is the asset critical?
- Are there known business risks associated with this asset?
- Do we have a network diagram to provide?
- What functions or features are most important to our organization, and do they require special attention?

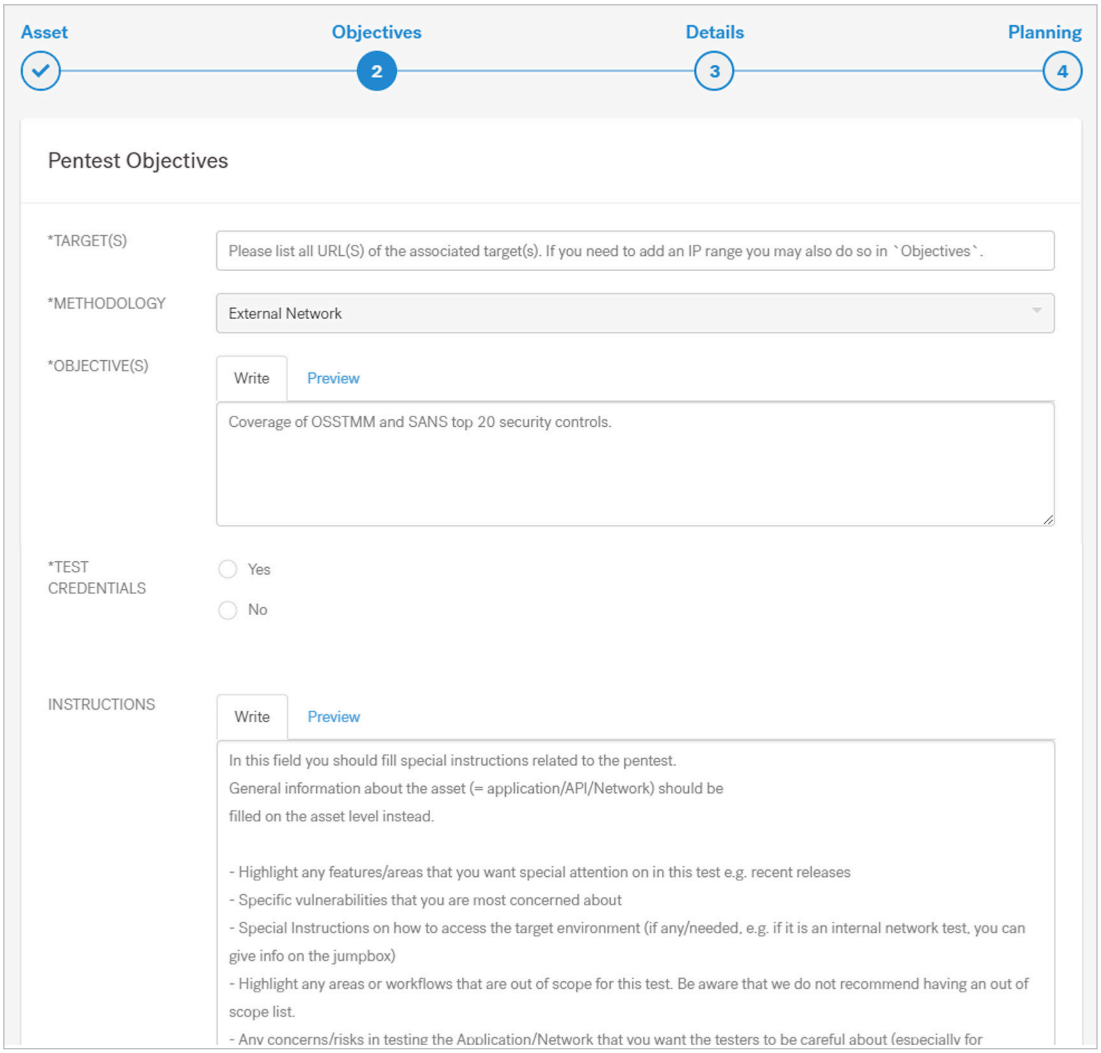
These questions are critical in helping define one's environment, but they are not always asked by internal teams. While trying to glean information about our environment, Cobalt was simultaneously making us consider what we had in our environment. We loved the introspection and would encourage organizations to spend time correctly classifying and labeling their assets. There's another hidden gem here: the ability to easily *repeat* a pentest with *consistency*. Once an asset is defined, it's *defined*. You can schedule multiple tests against it over a period of time and track the outcome of those tests (more on this later). You can change the context of an asset with ease, by simply modifying text or dragging and dropping new associated items.

> One of our favorite features from Cobalt is the ability to *schedule a pentest on demand, when we saw fit*. Within a matter of minutes, we could add an asset and schedule a test against it, allowing us to address business risks in a matter of moments.

Keeping in line with its minimalistic design, once an asset has been classified in Cobalt's portal, we could begin to test against it immediately. Herein lies another benefit we identified during our review. We were able to add an asset and schedule a test within *minutes*, a time frame that typically takes days and weeks. We liked the time savings, but also knew that when we scheduled a test, the skilled Cobalt team was receiving all the context we provided to make the test a successful one. At this stage, you can also add collaborators to a pentest. Initially, we viewed this feature as a way to include others from the security team. However, as we thought through this further, this is also a unique way to include others from *outside* the security team. For example, consider an application developer who may be interested in the security risks of their code. Include them in the pentest, and let them observe the results in real time!

Time to Test

With assets defined, scheduling a penetration test is a quick and easy process. As shown in Figure 5, scheduling a test is yet another chance to define what you hope to achieve from the penetration test.



Pentest Objectives

*TARGET(S)	Please list all URL(S) of the associated target(s). If you need to add an IP range you may also do so in `Objectives`.
*METHODOLOGY	External Network
*OBJECTIVE(S)	Write Preview Coverage of OSSTMM and SANS top 20 security controls.
*TEST CREDENTIALS	○ Yes ○ No
INSTRUCTIONS	Write Preview

In this field you should fill special instructions related to the pentest.
General information about the asset (= application/API/Network) should be
filled on the asset level instead.

- Highlight any features/areas that you want special attention on in this test e.g. recent releases
- Specific vulnerabilities that you are most concerned about
- Special Instructions on how to access the target environment (if any/needed, e.g. if it is an internal network test, you can give info on the jumpbox)
- Highlight any areas or workflows that are out of scope for this test. Be aware that we do not recommend having an out of scope list.
- Any concerns/risks in testing the Application/Network that you want the testers to be careful about (especially for

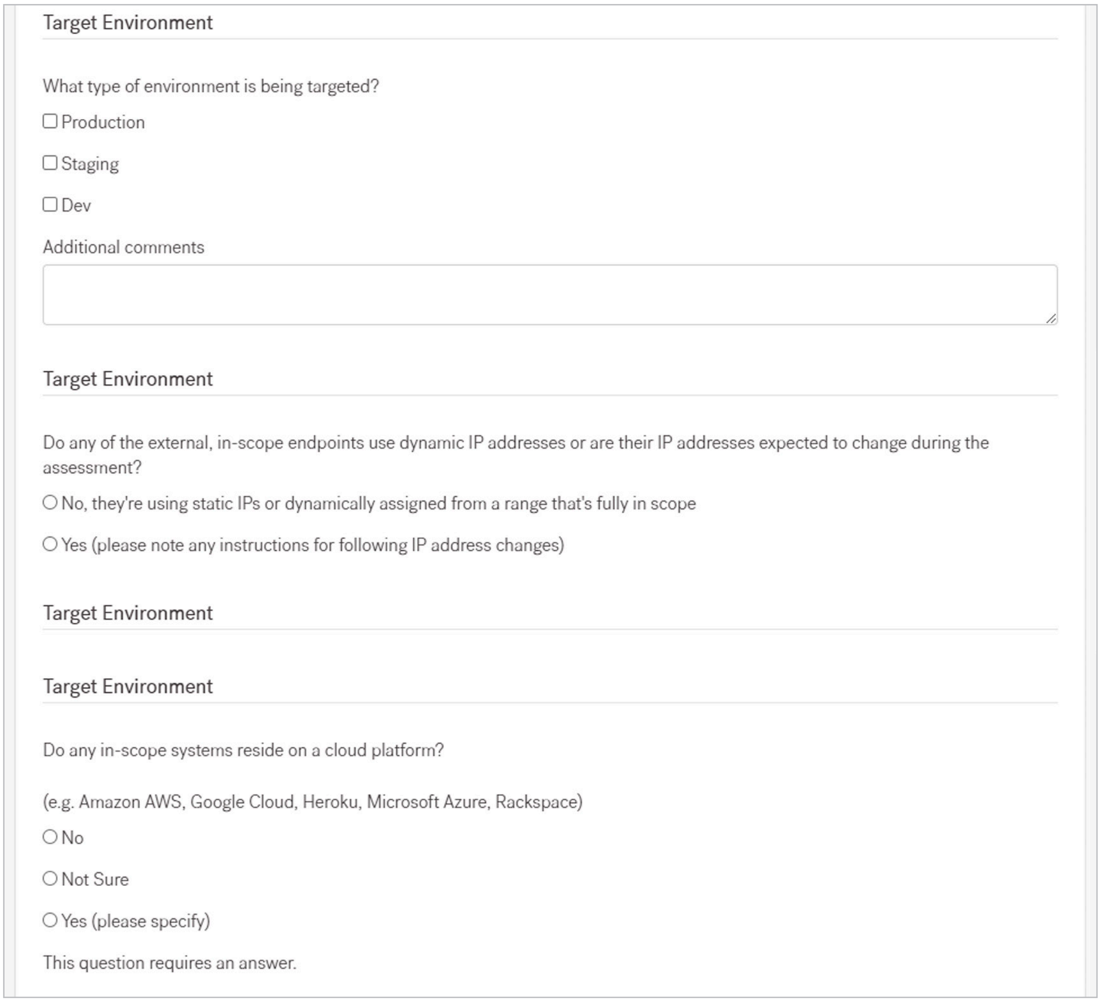
Figure 5. Identifying Pentest Objectives

As shown in Figure 5, there are some key details that Cobalt is hoping to capture:

- **Target(s)—**What are the targets of the penetration test? These may be defined at a high-level as the asset.
- **Objective(s)—**What do you hope to achieve from the test? Is it a simple vulnerability check, or do you want to test for the Top 20 CIS Controls?[1]
- **Test credentials—**Would you like to provide credentials for the Cobalt team to use as part of their assessment? Depending on your objectives, you may wish to provide credentials ahead of time.
- **Instructions—**This section, perhaps the lengthiest portion of Objectives, asks you to specify key details about the asset, the type of test you're requesting, and any other items you want included in scope.

These details are straightforward for any penetration test. As we have previously stated, there are multiple points in the Cobalt platform where a customer is meant to *think deeper* about their environment and what they hope to achieve from a penetration test. When scheduling the test for our review, we found ourselves evaluating our goals for conducting the test and defined our objectives clearly. The more information provided, the better the Cobalt experience will be.

Once you have submitted details, the web form will do a quick sanity check against the information you provided and the data points available for your organization. Cobalt will also ask some follow-up technical questions, shown in Figure 6.



Target Environment

What type of environment is being targeted?

□ Production

□ Staging

□ Dev

Additional comments

Target Environment

Do any of the external, in-scope endpoints use dynamic IP addresses or are their IP addresses expected to change during the assessment?

○ No, they're using static IPs or dynamically assigned from a range that's fully in scope

○ Yes (please note any instructions for following IP address changes)

Target Environment

Target Environment

Do any in-scope systems reside on a cloud platform?

(e.g. Amazon AWS, Google Cloud, Heroku, Microsoft Azure, Rackspace)

○ No

○ Not Sure

○ Yes (please specify)

This question requires an answer.

Figure 6. Additional Technical Questions

[1] www.cisecurity.org/controls

These questions are there to ensure consistency between test scheduling and execution. Networked infrastructure can change quickly, and capturing static or dynamic details helps ensure the test will be run against the correct asset. We ran a penetration test against a cloud-hosted virtual machine, for example, and had to specifically assign a static IP so it would not change daily.

Let us pause and consider some hidden value within Cobalt's platform. At this point, we've inserted multiple data points, fully defined an asset, and planned a penetration test. The platform not only made this process easy, but data is also stored for posterity. This is one of the best features of Cobalt: the ability to save and repeat tests for weeks, months, or years to come. One of the biggest inconsistencies we see in the industry is that tests may not be cognizant of previous tests, essentially reinventing the wheel each time. Cobalt eliminates this problem, and we loved it.

Once objectives are defined, descriptions provided, and technical details sorted, your pentest request goes into Cobalt's scheduling system. Your dashboard will be updated to reflect a pending penetration test, and Cobalt will provide a separate dashboard board to assist in tracking the test from beginning to end. Thus, the Cobalt experience begins.

The Cobalt Experience

Once you have scheduled a penetration test against an asset, Cobalt gets multiple wheels in motion. Designed to optimize the test you have scheduled, these simultaneous activities focus on skill alignment and communication and include the following:

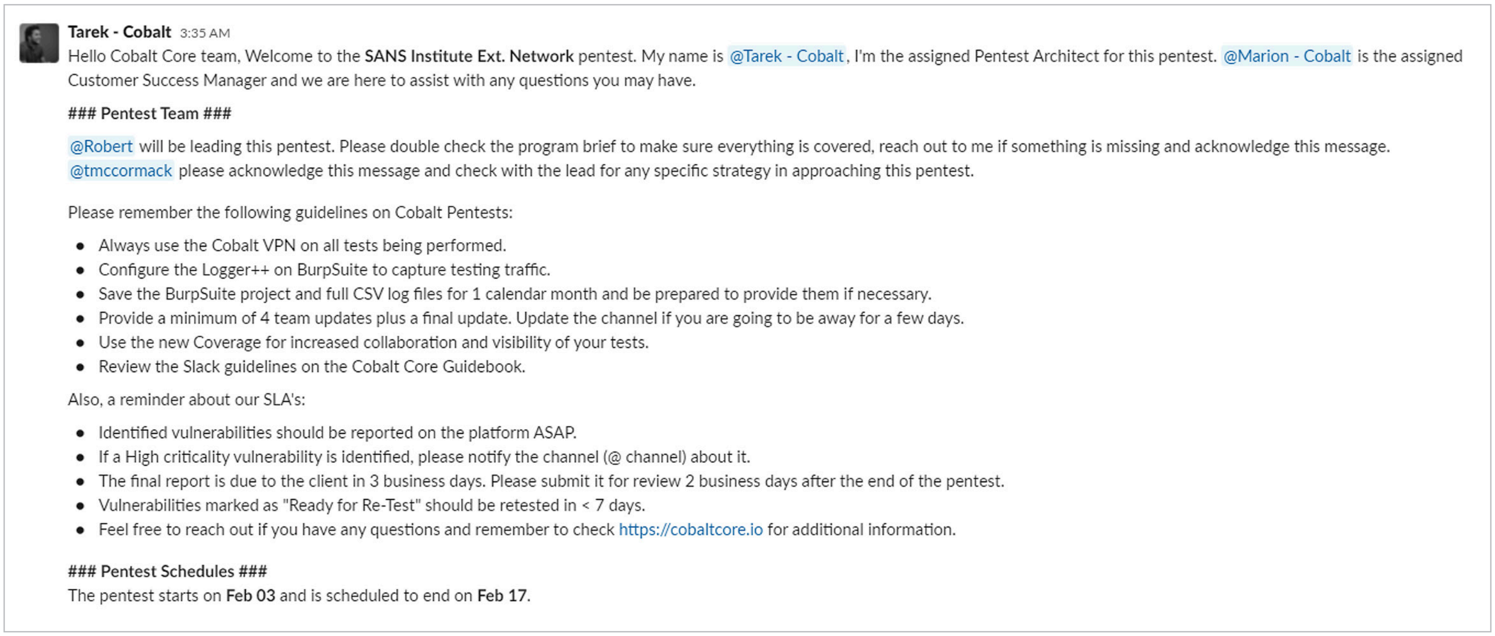
- Cobalt begins to find the right resources who are skilled for the test you requested and align their schedule with yours.
- Cobalt sets up a Slack channel that will ultimately house the parties you specified, the skilled penetration testers assigned to your test, and a Cobalt customer success manager.

Cobalt provides a unique skills advantage because they can use penetration testers of all skill sets and those with remote availability to meet the various objectives and assets of their customers.

Any issues identified with the scheduling of your test will be called out and rectified in the Slack channel. For example, when describing our first test, we inadvertently included two incompatible assets in the same test. The Cobalt representative we worked with noticed this and offered to correct it immediately, ensuring our scope was not too unwieldy. We loved the instantaneous feedback!

Real-Time Interaction

Once all details have been resolved, the PM team will schedule your test with the correct resources. As shown in Figure 7, we were assigned a lead, and our penetration test was scheduled from February 3–17, 2021.



Tarek - Cobalt 3:35 AM
Hello Cobalt Core team, Welcome to the **SANS Institute Ext. Network** pentest. My name is @Tarek - Cobalt, I'm the assigned Pentest Architect for this pentest. @Marion - Cobalt is the assigned Customer Success Manager and we are here to assist with any questions you may have.

Pentest Team

@Robert will be leading this pentest. Please double check the program brief to make sure everything is covered, reach out to me if something is missing and acknowledge this message.
@tmccormack please acknowledge this message and check with the lead for any specific strategy in approaching this pentest.

Please remember the following guidelines on Cobalt Pentests:

- Always use the Cobalt VPN on all tests being performed.
- Configure the Logger++ on BurpSuite to capture testing traffic.
- Save the BurpSuite project and full CSV log files for 1 calendar month and be prepared to provide them if necessary.
- Provide a minimum of 4 team updates plus a final update. Update the channel if you are going to be away for a few days.
- Use the new Coverage for increased collaboration and visibility of your tests.
- Review the Slack guidelines on the Cobalt Core Guidebook.

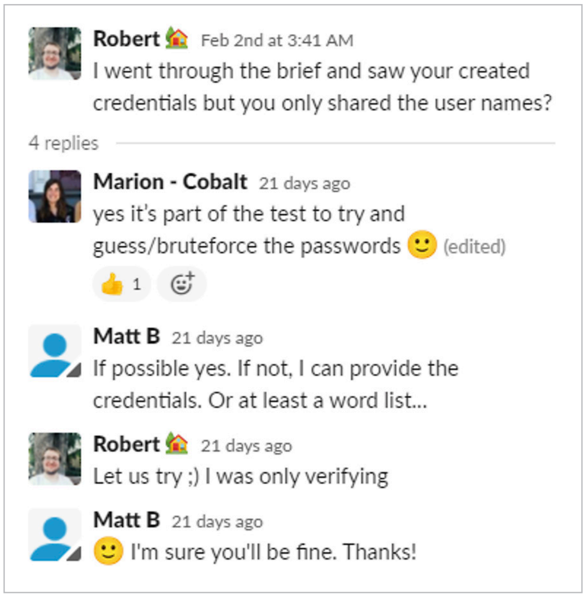
Also, a reminder about our SLA's:

- Identified vulnerabilities should be reported on the platform ASAP.
- If a High criticality vulnerability is identified, please notify the channel (@ channel) about it.
- The final report is due to the client in 3 business days. Please submit it for review 2 business days after the end of the pentest.
- Vulnerabilities marked as "Ready for Re-Test" should be retested in < 7 days.
- Feel free to reach out if you have any questions and remember to check https://cobaltcore.io for additional information.

Pentest Schedules
The pentest starts on **Feb 03** and is scheduled to end on **Feb 17**.

Figure 7. Slack Channel for Our Penetration Test

You may notice in Figure 7 that the "welcome message" provides guidelines for both penetration testers *and* customers, which is great for holding the team accountable for their findings and ensuring the test is completed in a timely manner. Furthermore, once the test has been scheduled and resources assigned, Slack becomes the primary method of communication for the penetration test. The customer success team checked in with us periodically, while the testers themselves used the channel to facilitate discussions on hypotheses, findings, and questions. For example, Figure 8 provides a snippet of a conversation where the penetration testers identified that we did not provide credentials, confirming that password brute forcing was part of our objectives.



Robert 🏡 Feb 2nd at 3:41 AM
I went through the brief and saw your created credentials but you only shared the user names?

4 replies

Marion - Cobalt 21 days ago
yes it's part of the test to try and guess/bruteforce the passwords 🙂 (edited)
👍 1

Matt B 21 days ago
If possible yes. If not, I can provide the credentials. Or at least a word list...

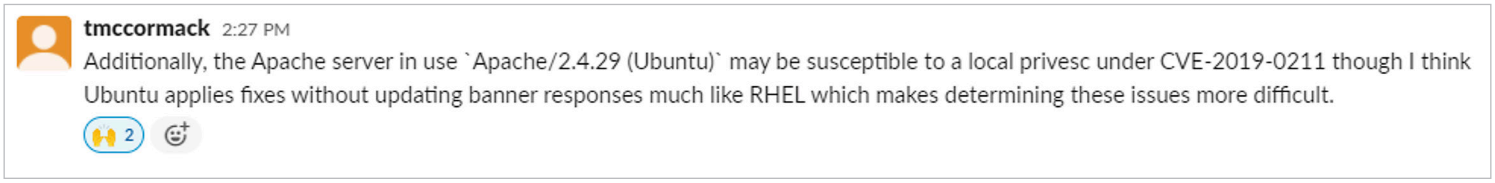
Robert 🏡 21 days ago
Let us try ;) I was only verifying

Matt B 21 days ago
🙂 I'm sure you'll be fine. Thanks!

Figure 8. Conversation Between SANS and Cobalt Regarding Our Pentest

Cobalt's use of Slack as a real-time communication medium gave us a front-row seat to the penetration test as it was conducted in our environment. We were able to interact with the testers directly during the test, addressing any roadblocks (whether intended or not).

As the penetration test progressed, the channel became a window into the test itself. As shown in Figure 9, the testers placed forth a hypothesis on an Apache vulnerability.



tmccormack 2:27 PM

Additionally, the Apache server in use `Apache/2.4.29 (Ubuntu)` may be susceptible to a local privesc under CVE-2019-0211 though I think Ubuntu applies fixes without updating banner responses much like RHEL which makes determining these issues more difficult.

Figure 9. Snippet from Our Channel, Outlining a Hypothesis for a Vulnerable Web Server

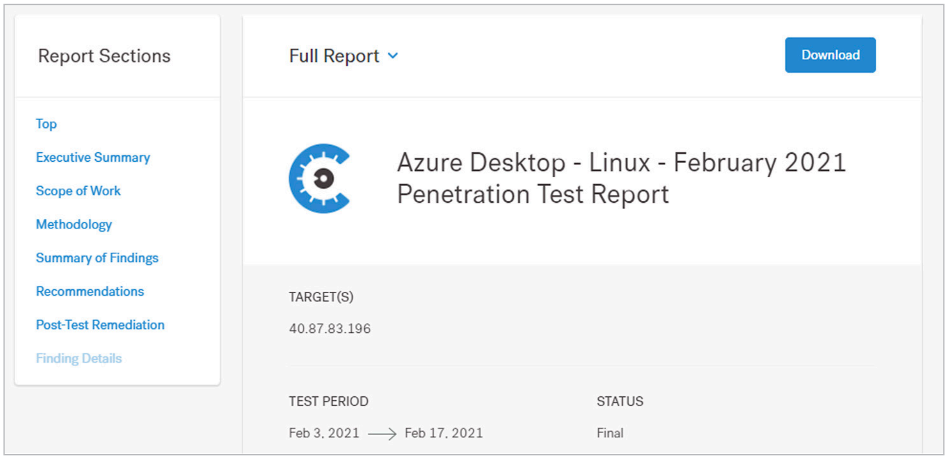
It is within the Slack channel that the true idea of the Cobalt experience is realized. In traditional situations, penetration tests are behind-the-scenes affairs, with many organizations simply waiting until a final report. Our issue with this approach, despite being the industry norm, is that organizations are left in the dark, wondering if their defenses even worked. Worse still, a two-week silence can lead to an embarrassing penetration report, shaming defenses and making defenders squirm for a lack of detection. Neither situation allows the organization to mature.

With our front-row seat into the penetration test, we were able to observe the approach that the testers took to break into our environment. Were they successful in every attempt? Certainly not. But as defenders, we were able to glean the *thought process* behind a particular hypothesis and identify its successes and failures. Our team was able to gain immediate knowledge of what works and what does not, and that truly shaped our thinking on security better than any final report could.

Toward the end of the penetration test, the Cobalt team performed multiple checks on scheduling and confirmed that they had completed all objectives. Considering our requirements were broad (we essentially asked the team to try and break in and see what they could do), they met these objectives. The customer success team also kept a watchful eye on the entire process and followed up once the report was published.

The Report

The interactive testing process was unique and insightful, and something we highly recommend. While we understand that not every organization can allow the entire blue team to interact in the channel, we found extreme value in interacting with the testers themselves. For team members who do not partake in the live channel, the final report becomes their method of consumption.[2]



Azure Desktop - Linux - February 2021 Penetration Test Report

TARGET(S)
40.87.83.196

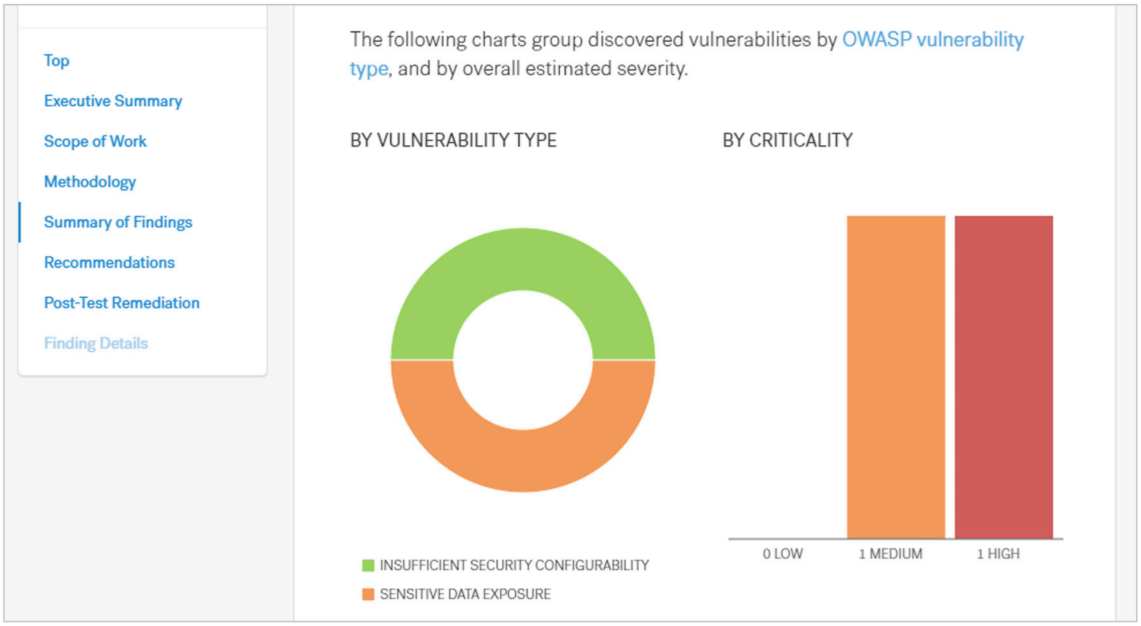
TEST PERIOD
Feb 3, 2021 → Feb 17, 2021

STATUS
Final

Figure 10. Top of Our Penetration Test Report

Depending on your level of involvement in the channel and with the testers, you may already be aware of the contents of the report. As shown in Figure 10, the report is broken down into easily navigable sections.

[2] For final report consumption, Cobalt's platform lends very well to displaying the results of our penetration test. There is also a downloadable PDF version, which is well formatted and provides the same level of detail. For the purposes of our product review, our screenshots are taken from the report in the browser.



The following charts group discovered vulnerabilities by OWASP vulnerability type, and by overall estimated severity.

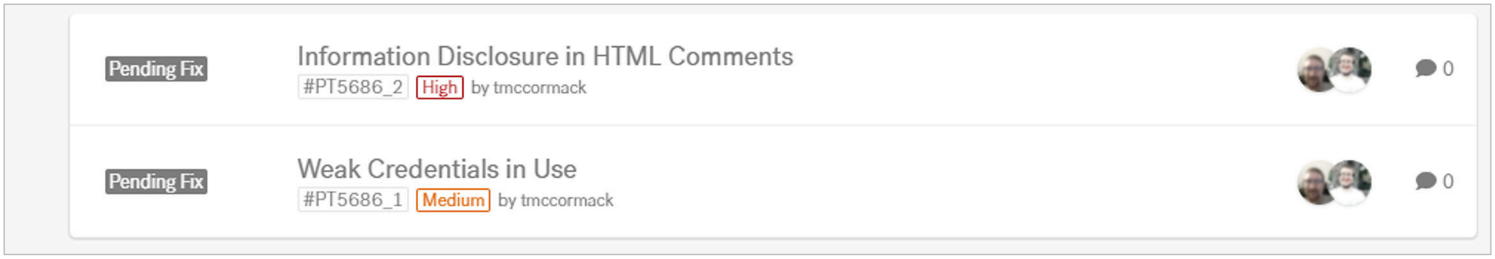
BY VULNERABILITY TYPE
BY CRITICALITY
0 LOW
1 MEDIUM
1 HIGH
🟩 INSUFFICIENT SECURITY CONFIGURABILITY
🟧 SENSITIVE DATA EXPOSURE

Figure 11. Summary of Findings from Our Penetration Test Report

With the scope of work and methodology clearly outlined during the scheduling of the test, we navigated to the Summary of Findings section to discover what the testers identified. Because our asset was a simple virtual machine with little footprint and/or functionality, we did not expect findings to be critical or require immediate attention. These famous last words have cost many organizations. Even a simple system with weak credentials can be an easy entry vector into an organization.

Our test resulted in two findings: one medium and one high. As shown in Figure 11, our system had sensitive data exposure and an insufficient security configuration.

Of course, many organizations want more than simple high-level findings. Luckily, within the same navigable report format, each finding is called out with its own data points. Figure 12 shows an example of the Findings tab and the two findings from our penetration test.

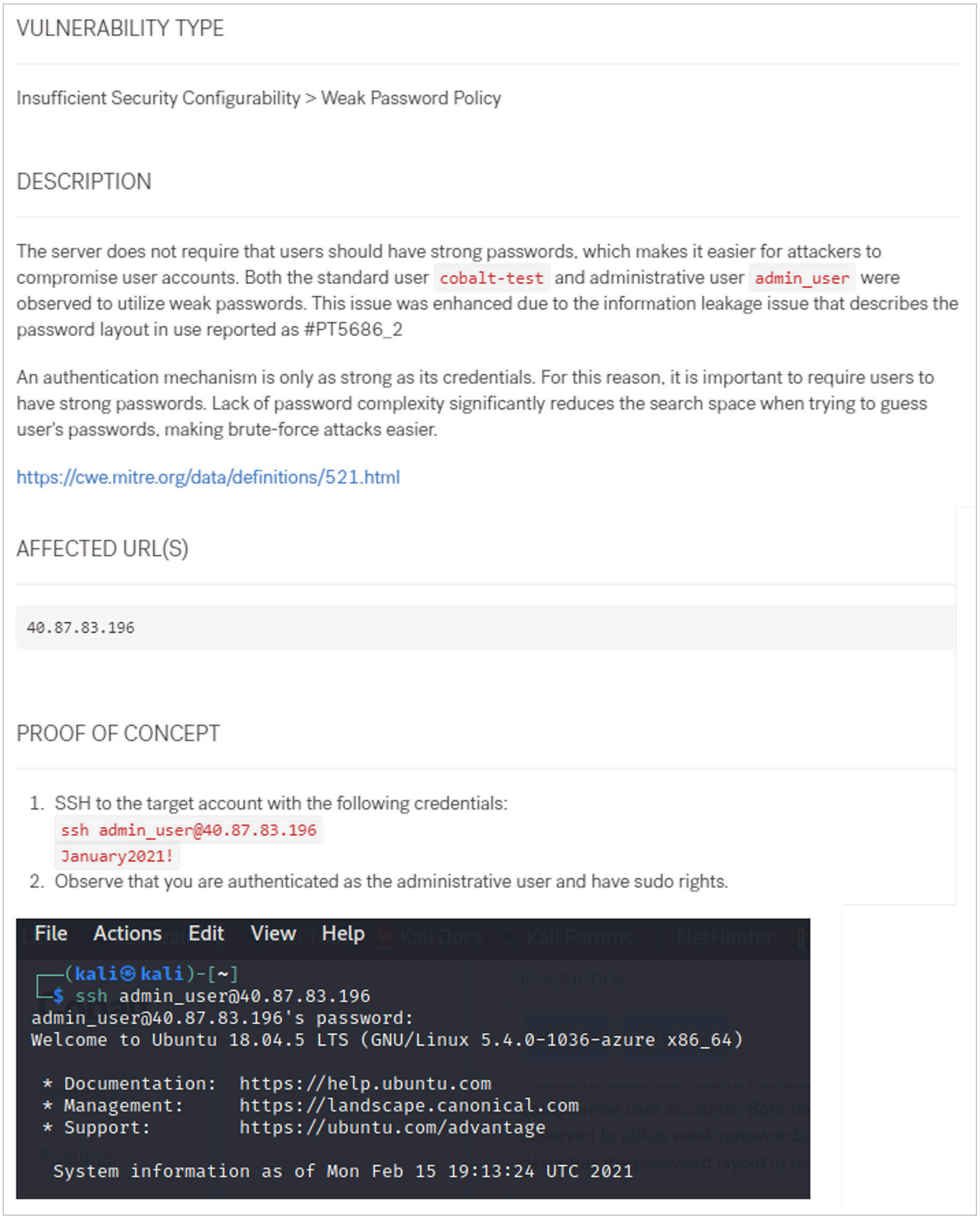


Pending Fix — Information Disclosure in HTML Comments
#PT5686_2 High by tmccormack — 💬 0

Pending Fix — Weak Credentials in Use
#PT5686_1 Medium by tmccormack — 💬 0

Figure 12. Findings Tab from Our Penetration Test

From the Findings tab, we can drill down deeper into the technical details of each. For example, in their Medium risk finding, what did the Cobalt testers mean by “Weak Credentials in Use”? Cobalt thought ahead and ensures that each finding reported in the test is properly documented with examples and a proof of concept. Figure 13, on the next page, shows the technical details surrounding this finding.



VULNERABILITY TYPE

Insufficient Security Configurability > Weak Password Policy

DESCRIPTION

The server does not require that users should have strong passwords, which makes it easier for attackers to compromise user accounts. Both the standard user `cobalt-test` and administrative user `admin_user` were observed to utilize weak passwords. This issue was enhanced due to the information leakage issue that describes the password layout in use reported as #PT5686_2

An authentication mechanism is only as strong as its credentials. For this reason, it is important to require users to have strong passwords. Lack of password complexity significantly reduces the search space when trying to guess user's passwords, making brute-force attacks easier.

https://cwe.mitre.org/data/definitions/521.html

AFFECTED URL(S)

```
40.87.83.196
```

PROOF OF CONCEPT

1. SSH to the target account with the following credentials:
   `ssh admin_user@40.87.83.196`
   `January2021!`
2. Observe that you are authenticated as the administrative user and have sudo rights.

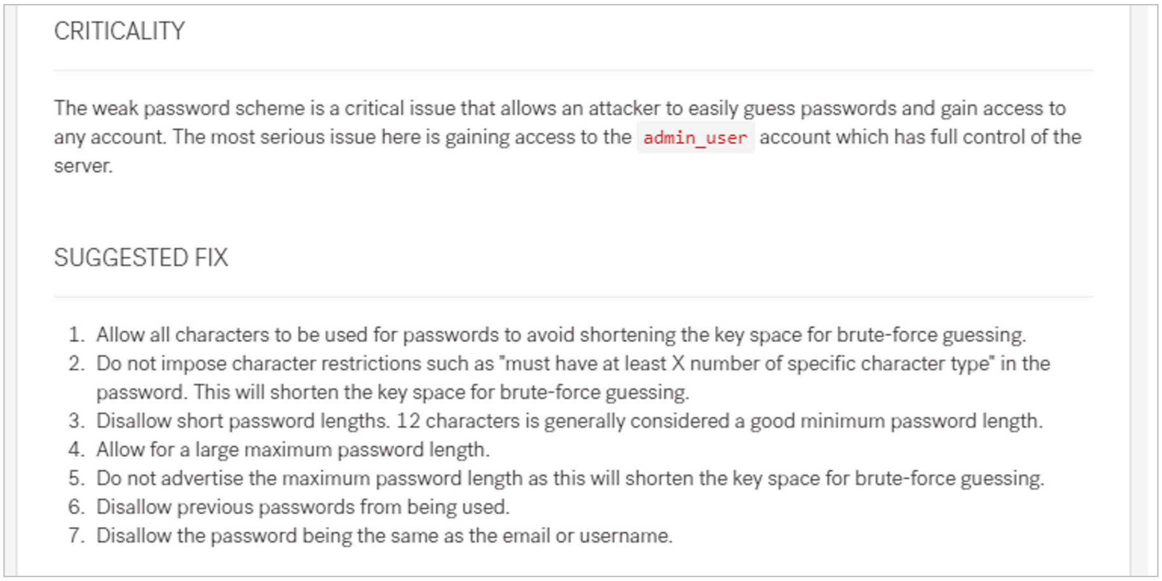
```
┌──(kali㉿kali)-[~]
└─$ ssh admin_user@40.87.83.196
admin_user@40.87.83.196's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-1036-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon Feb 15 19:13:24 UTC 2021
```

Figure 13. Report: Identifying Weak Credentials on Our Test System

Of course, simply providing a security weakness isn't enough. Cobalt ensures that each finding is also paired with a definition of its criticality *and* a list of recommendations to mitigate the factor. Figure 14 continues the screenshot from Figure 13 for the finding of weak credentials.



CRITICALITY

The weak password scheme is a critical issue that allows an attacker to easily guess passwords and gain access to any account. The most serious issue here is gaining access to the `admin_user` account which has full control of the server.

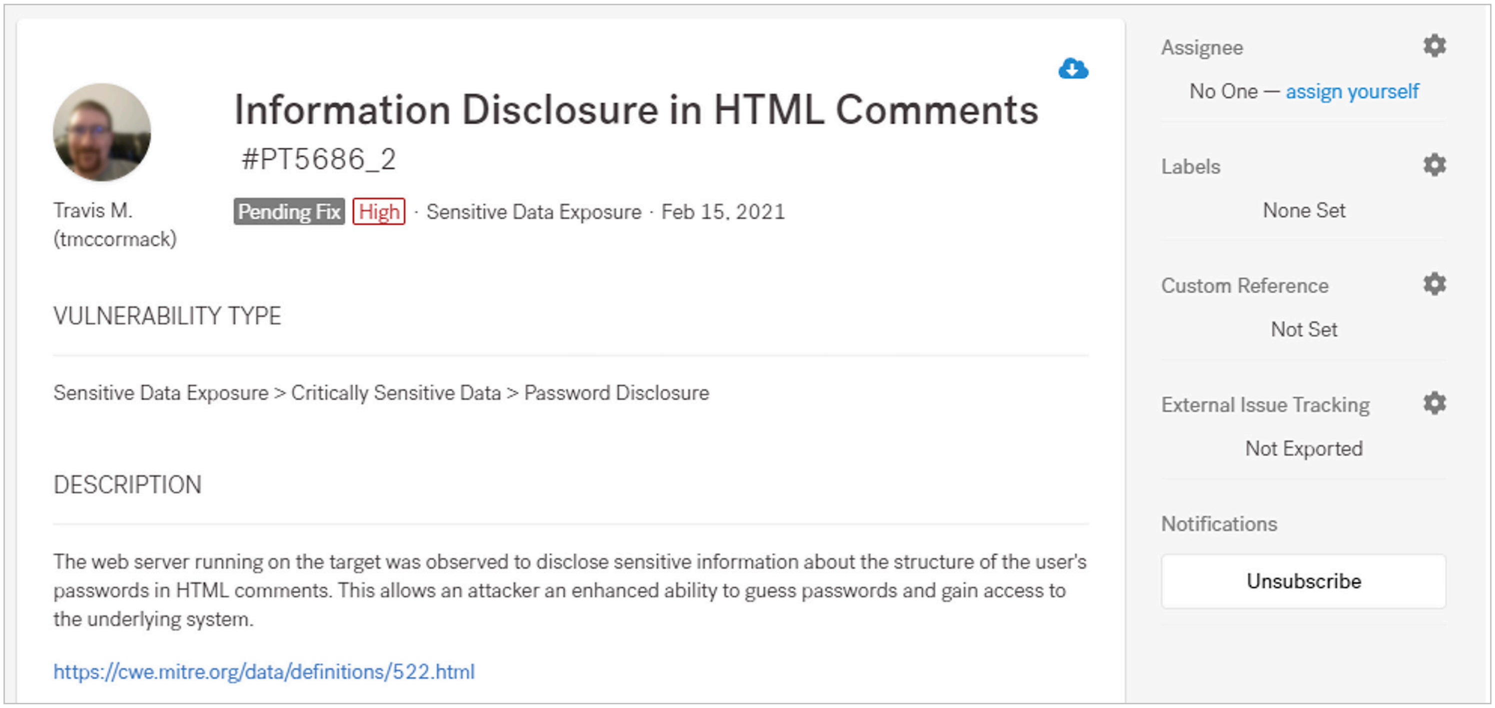
SUGGESTED FIX

1. Allow all characters to be used for passwords to avoid shortening the key space for brute-force guessing.
2. Do not impose character restrictions such as "must have at least X number of specific character type" in the password. This will shorten the key space for brute-force guessing.
3. Disallow short password lengths. 12 characters is generally considered a good minimum password length.
4. Allow for a large maximum password length.
5. Do not advertise the maximum password length as this will shorten the key space for brute-force guessing.
6. Disallow previous passwords from being used.
7. Disallow the password being the same as the email or username.

Figure 14. Report: Criticality and Recommendations for a Finding

It's here that many organizations are used to stopping. The report has been published and findings received. It is now time for the security team to go fix the weaknesses. We are done, right? Absolutely not! Cobalt again thought ahead and decided that accountability is the best way to ensure fixes get resolved. Herein lies yet another great feature of the Cobalt platform: issue tracking integration.

Let us look at another screenshot, this time for the finding of "Information Disclosure in HTML Comments" (see Figure 15).



Travis M.
(tmccormack)

Information Disclosure in HTML Comments
#PT5686_2

Pending Fix High · Sensitive Data Exposure · Feb 15, 2021

VULNERABILITY TYPE

Sensitive Data Exposure > Critically Sensitive Data > Password Disclosure

DESCRIPTION

The web server running on the target was observed to disclose sensitive information about the structure of the user's passwords in HTML comments. This allows an attacker an enhanced ability to guess passwords and gain access to the underlying system.

https://cwe.mitre.org/data/definitions/522.html

Assignee
No One — assign yourself

Labels
None Set

Custom Reference
Not Set

External Issue Tracking
Not Exported

Notifications
Unsubscribe

Figure 15. Report: Information Disclosure in HTML Comments Finding

Looking at the right-hand side of Figure 15, some of our readers might recognize an issue tracking system—Cobalt uses JIRA and GitHub out of the box. The findings we have examined in the previous handful of figures are issues, defined within the web platform as "fixes" that need to be implemented. You will notice in Figure 15 that you can assign the issue to someone, apply labels, and add commentary to the end of the issue.

This is an extremely valuable feature and integration. By immediately taking penetration testing findings and converting them to issues, Cobalt established accountability to ensure that we fixed our weaknesses. Furthermore, *included in the penetration test*, is the ability to retest a weakness or vulnerability. Again, we see a theme of repeatability and consistency. If your team says that they fixed something, the Cobalt pentesters will verify and report back, allowing you to close an issue with confidence that the weakness has been remediated.

Conclusion

Overall, working with Cobalt was a highly positive experience. It is worth noting that we did everything from within our browser, requiring only two tabs (one for the Cobalt platform, the other for Slack). This streamlining of processes and communications made us fall in love with Cobalt's approach to on-demand penetration testing. We had complete control over asset classification and test scheduling. Furthermore, the test data is saved—meaning we can repeat tests with consistency as many times as we need to.

We maintained front row visibility with the testing team, allowing us to peek into the minds of our testers. The final report delivered impactful takeaways and key insights into business risk, coupled with an issue-tracking system that ensured gaps were fixed. Even better, we can repeat the test against controls to ensure that the fixes took place.

Security control testing is necessary for any organization to ensure it maintains current defenses against well-known threats. Scheduling a penetration test and forgetting about it until you receive a final report is hardly an effective technique for testing your controls. Given today's dispersed workforce, with assets and security teams stretched across the globe, anywhere an organization can find efficiencies while maintaining security is welcome. Cobalt offers this and more by thinking of penetration tests as an *experience* that should be shared by both red and blue teams, helping each other learn and grow each step of the way.

About the Author

Matt Bromiley is a SANS digital forensics and incident response instructor, teaching FOR508: Advanced Incident Response, Threat Hunting, and Digital Forensics and FOR572: Advanced Network Forensics: Threat Hunting, Analysis, and Incident Response. He is a principal consultant at a global incident response and forensic analysis company, combining his experience in digital forensics, log analytics, and incident response and management. His skills include disk, database, memory and network forensics; incident management; threat intelligence; and network security monitoring. Matt has worked with organizations of all shapes and sizes, from multinational conglomerates to small, regional shops. He is passionate about learning, teaching and working on open source tools.

Sponsor

SANS would like to thank this paper's sponsor:

Cobalt
Pentest as a Service