



# COOL PENTESTING STORIES

*and Other Offensive Security Research*



# COOL PENTESTING STORIES

---

NetSPI is home to several of the world's top offensive security experts. In this booklet, you'll find six real stories from the NetSPI pentesting team, featuring our critical discoveries, unique approaches, and the impact we've made on some of the world's most prominent organizations.



## **CHAPTER 1**

**Full Source Code in a Healthcare Application**

## **CHAPTER 2**

**Using Basic Physics to Access the Silkscreen of a Printed Circuit Board**

## **CHAPTER 3**

**App Registration Certificates Stored in Azure Active Directory – CredManifest CVE**

## **CHAPTER 4**

**Bypassing a Voice Biometrics System Using Deepfakes**

## **CHAPTER 5**

**Not Your Average Bug Bounty – On-Site Social Engineering**

## **CHAPTER 6**

**Kerberos Bronze Bit Attack**

# Full Source Code in a Healthcare Application

## PROCESS: Attack Surface Management (ASM)



### Eric Gruber

Director, Attack Surface Management

EXPLORE ASM



---

## DISCOVERY & IMPACT

Sensitive information vulnerabilities do not always pose a direct threat to an organization's infrastructure, but those that do are handled with special urgency. In this instance, **the information disclosure discovered by ASM was as bad as they get**, exposing the full source code for the application, including configuration credentials and a full database backup containing user data and credentials for users of a healthcare-focused application.

In addition to threatening users' privacy by exposing their personal information to unauthorized parties, **this information disclosure also could enable an attacker to take full control over the application's infrastructure**, potentially leading to further information disclosures, deployment of ransomware, or any number of costly and damaging attacks.





## HOW WE DID IT

- 1 Upon detecting a potential information disclosure vulnerability, ASM alerted offensive security professionals on NetSPI's ASM Operations Team.
- 2 We then reviewed the vulnerability details and began the process of manual verification.
- 3 Manual review of the vulnerable location within the customer's application revealed a large archive file.
- 4 We downloaded and extracted the archive file to inspect its contents. Upon inspection, we discovered that the archive contained a complete backup of the entire web application, including all user data, hashed user passwords, and source code.
- 5 Though passwords were not being stored in plaintext, hashed passwords can sometimes be recovered to their plaintext form through the process of password cracking. To further assess the impact of the vulnerability, we extracted and successfully cracked a hashed user password from the database backup.
- 6 Using the cracked password and associated username, we were then able to log into the application, demonstrating that the database backup contained valid data.
- 7 Given the immediate risk posed to the customer's infrastructure, application, and its users' data, offensive security experts elevated the original Medium severity information disclosure vulnerability to a "Critical" severity rating and immediately reported the vulnerability to the customer, in line with their vulnerability reporting preferences.

## REMEDIATION

Though the potential impact of this vulnerability was severe, the fix was simple. Working with the customer, offensive security experts provided instructions on how to address the root cause of this vulnerability and to prevent similar vulnerabilities from arising in the future.



# Using Basic Physics to Access the Silkscreen of a Printed Circuit Board

## PROCESS: Embedded Penetration Testing



### Shay Galland

Security Consultant II - IoT/Embedded

LEARN MORE ABOUT  
EMBEDDED PENTESTING



---

## DISCOVERY & IMPACT

Our client wanted to research whether a circuit board coated in epoxy potting is an effective security measure to protect against reverse engineering attacks on printed circuit boards. The goal in this research was to be able to see silkscreen markings and map out component locations on the circuit board. We used a temperature attack based on the simple physics principle that materials will shrink at different rates when they are cooled past a certain degree. In this case, the fiberglass PCB will shrink at a different rate than epoxy, thus creating a separation between the two. By using liquid nitrogen, we were able to weaken the adhesion of the epoxy and eventually separate it from the circuit board, giving access to the (mirrored) silkscreen embedded in the epoxy.





## HOW WE DID IT

- 1 Acquire liquid nitrogen from a local welding supply store.
- 2 The liquid nitrogen made the epoxy's adhesion to the board more brittle, so we had to work delicately applying enough force to widen the crack and allow more nitrogen in but not so much force as to break the epoxy.
- 3 Little by little we were able to use scraping and pry tools to separate the epoxy from the circuit board into two pieces.
- 4 Almost the entire silkscreen layer came off of the PCB, but with the help of a black light and a bit of sandpaper we were able to see the (mirrored) silkscreen embedded in the epoxy.
- 5 Many/most of the components were ripped off the PCB, but the solder pads were still there, and we were able to trace the pins of the microcontroller to the debug headers.
- 6 Ultimately, we were able to show that even if the de-potted sample was destroyed in the process of removing the epoxy that an attacker would be able to see information such as silkscreen markings, general circuit layout, and the details and locations of critical components.

## REMIEDIATION

The conclusion of this research is that epoxy alone is not a sufficient security control against reverse engineering attacks. Appropriate security controls would be to remove any unnecessarily informational silkscreen markings such as those indicating critical components, ensure that critical components such as microcontrollers and flash have read and write protections enabled, encrypt firmware, and add a layer of resistive foil or similar as a tamper measure to ensure that the device won't function if an attacker drills through the epoxy to access components.



# CredManifest: App Registration Certificates Stored in Azure Active Directory

**PROCESS:** Cloud Penetration Testing



**Karl Fosaaen**

Senior Director

**READ THE FULL BLOG**



---

## **DISCOVERY & IMPACT**

Karl identified a misconfiguration in Azure where Automation Account “Run as” credentials were stored in cleartext in Azure Active Directory (AAD). This resulted in an impactful privilege escalation, as it would allow any user in this environment to escalate to Contributor of any subscription with an Automation Account.





## HOW WE DID IT

- 1 Identified an issue in the way the Automation Account “Run as” credentials were created when creating a new Automation Account in Azure.
- 2 Manually extract credentials by copying the certificate data out of the manifest and converting it to a PFX file. We did this with two lines of PowerShell.
- 3 Import the certificate to our local store using PowerShell in a local administrator session.
- 4 Use the newly installed certificate to authenticate to the Azure subscription as the App Registration.
- 5 With the Directory (Tenant) ID, App (Client) ID, and Certificate Thumbprint values available, run the Add-AzAccount command to authenticate to the tenant.
- 6 Develop PowerShell script to automate extraction.

## REMIEDIATION

- Karl responsibly disclosed the vulnerability to Microsoft who has since deployed updates that prevent cleartext private key data from being stored during application creation and prevents access to private key data previously stored.
  - NetSPI recommends cycling existing Automation Account “Run as” certificates, given the potential exposure of these credentials.
- 

# Bypassing a Voice Biometrics System Using Deepfakes

**PROCESS:** Red Team



**Alex Poorman**

Principal Consultant

**READ THE FULL BLOG**



---

## **DISCOVERY & IMPACT**

We were tasked with bypassing a voice biometric system for a Fortune 500 company, which had defenses in place to detect deep fake voices. In the end, **we successfully bypassed their authentication system during a red team test using machine learning and deep fake research.**

The relative ease of this attack process combined with the constant advancing nature of deep fake research and machine learning programs leads us to believe that in the next decade, this technology will rapidly improve and attacks like this will become more common.





## HOW WE DID IT

- 1 Research various existing machine learning programs that allow deep fake creation. We used Coqui TTS.
- 2 Set up the program. This was a difficult and time-consuming task as the dependencies were incredibly inconsistent and often conflicted.
- 3 Alex then recorded his own voice for sample training data using Mimic Recording Studio.
- 4 Mimic Recording Studio contains 30,000 phrases that the recorder can say, many of which came from the Warren Commission. By providing the known text of an audio recording, Coqui TTS can learn how various words should sound.
- 5 Next, the training was configured and monitored to ensure it was learning effectively.  
  
Synthesize audio for the final deep fake attempt.
- 6 Setup the voice biometric system using Alex's real, legitimate voice. The phrase that we had to say was "Please authenticate me with my voice."
- 7 Begin the attack using the deep fake. Call the voice biometric system and when prompted to authenticate with my voice, play the deep fake. After a few seconds, the system successfully authenticated him!

## REMEDIATION

We presented our findings and advised the organization against using this form of authentication until they could retest and prevent illegitimate authentication.



# Not Your Average Bug Bounty

## PROCESS: On-Site Social Engineering



### Dalin McClellan

Senior Security Consultant

READ THE FULL BLOG



---

## DISCOVERY & IMPACT

During an on-site social engineering test, NetSPI successfully gained unauthorized access to a high-security datacenter. We identified and reported weaknesses in the on-site security policies and controls. We demonstrated how social engineering can be used to bypass even the most sophisticated physical and technical security controls.





## HOW WE DID IT

- 1 Pretext. Identify a believable reason for being on-site. After looking at an approved list of vendors, we identified a national pest control company.
- 2 We mocked up confirmation and scheduling emails that imitated the company, confirming an appointment for the next day. Then crafted an email chain using a lookalike domain to make it appear as if the communications were coming from someone internally.
- 3 We purchased screen printed shirts, rented a similar company vehicle, and purchased a static cling with the logo. All within two days and for less than \$150.
- 4 Given our prework, they were expecting us. Without question, they swiped their badge, scanned their retinas, and opened the doors for us.
- 5 Within minutes, we were on the datacenter floor, and even gained access to the ceilings with network cable access.
- 6 After the engagement, we wanted to push a little further given the access we were able to achieve. We went back in and attempted to print a document.
- 7 They gave us temporary network credentials and we convinced the contact to let us email him the attachment, which he opened and printed for us.
- 8 We then left the site undetected.

## REMEDIATION

We provided the client with recommendations on how to address the security gaps we observed while on-site. The main vulnerability we exploited on this test was the fact that procedures for scheduling and confirming vendor visits were poorly defined.



# Kerberos Bronze Bit Attack

## PROCESS: Network Penetration Testing



**Jake Karnes**

Managing Consultant

READ THE FULL BLOG



---

## DISCOVERY & IMPACT

While researching Kerberos in Active Directory, Jake found a way to bypass two protections for the authentication protocol. The Kerberos Bronze Bit attack would allow intruders to impersonate privileged users and access sensitive network services.





## HOW WE DID IT

- 1 Comprehensive research into Kerberos and Microsoft's extensions in Active Directory. Jake's research was built upon research documented by Elad Shamir titled, *Wagging the Dog: Abusing Resource-Based Constrained Delegation to Attack Active Directory*
- 2 Took a closer look at the TGS\_REP data structure returned by the KDC after the S4U2self exchange. Specifically, where the Forwardable flag is located in the response.
- 3 Jake found that the Forwardable flag was not in the Privileged Attribute Certificate (PAC). An attacker could decrypt, set the Forwardable flag's value to 1, and re-encrypt the service ticket.
- 4 Because it was not in the signed PAC, the KDC was unable to detect that the value was tampered with.
- 5 Successfully convert a non-forwardable ticket into a forwardable ticket.
- 6 This attack bypasses two key protections:
  - a The protection for TrustedToAuthForDelegation and the "Trust this computer for delegation to specified services only – Use Kerberos only" configuration.
  - b The protection for accounts which do not allow delegation.

## REMEDIATION

- Jake responsibly disclosed the vulnerability to Microsoft who has since released multiple patches for CVE-2020-17049.
  - The PAC now has an additional field which holds the "ticket signature" to detect tampering of tickets by parties other than the KDC.
- 



The Global Leader in Offensive Security

For more vulnerability  
research and penetration  
testing findings, visit:

[www.netspi.com/blog/technical](http://www.netspi.com/blog/technical)

---

Keep in Touch:



@netspi



@teamnetspi