HashiCorp

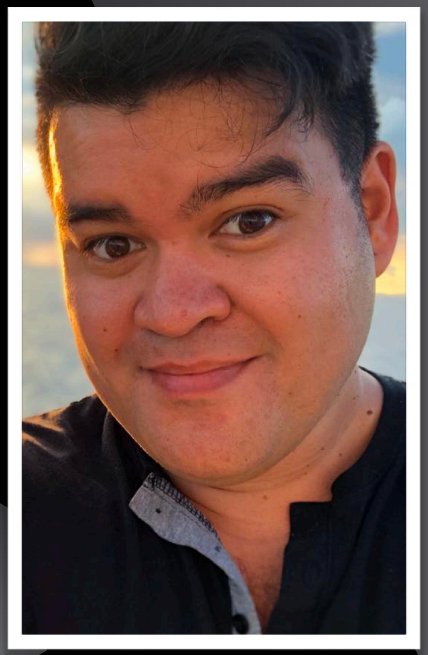
Defense Against the Dark Arts

Andy Manoske

Principal Product Manager at HashiCorp

Hi, I'm Andy!

(He/Him)



- PM: Cryptography and Security Products @ HashiCorp
- PM: Security Research @ AlienVault (Acquired by AT&T)
- PM: Encryption and Defense Systems @ NetApp
- Security and Enterprise Infrastructure VC @ Amplify Partners, GGV Capital

Tales from the Front

Vault's role in defending against adversaries in real data breaches

The Target

A major website using Vault.

What Happened?

Adversary breached perimeter security and TDE using advanced attack tool set.

Was Vault Breached?

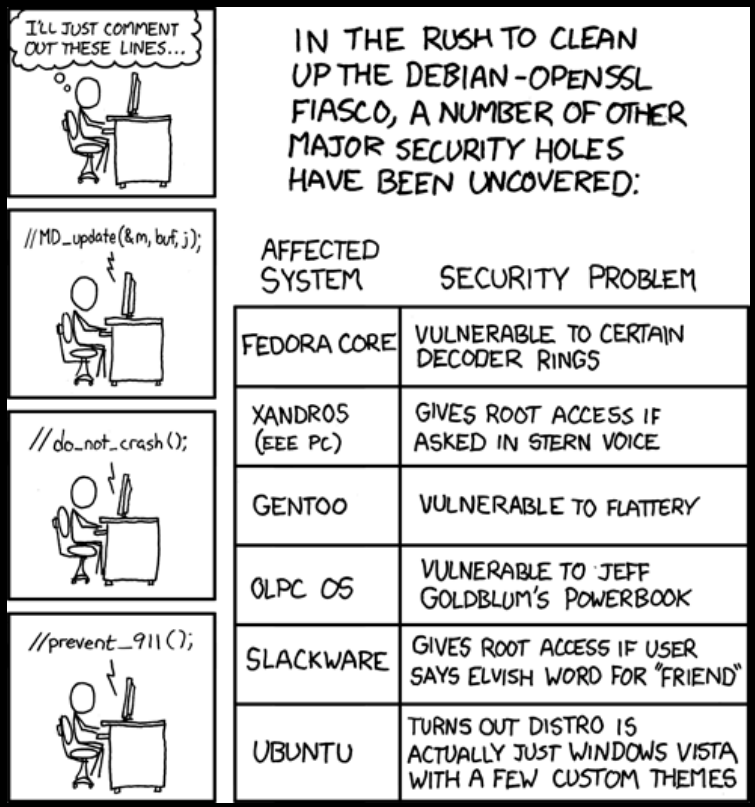
No. Adversaries were unable to decrypt data protected by Vault cryptography (keys stored in Vault were safe).

The Problem with Building Security Tools

Security means different things to different people

Whether something is “secure” depends on details like the following:

- Data Protected
- Compliance Requirements
- Threat Model



“What’s Secure Crypto Mean for You?”

Alexis Murphy - SRE at Jurassic Park

“It’s good enough to stand up to hackers and won’t fail me an audit.”

“John” (won’t give real name) - SRE at GovCloud

“I have to have post-quantum security capable of standing up to a 5000-qubit quantum computer performing cryptanalysis on it. Also it should be FIPS 140-3 certified to Level 2+. Also include duress passwords on login because people may try to kill me and I need a way to alert Fourth Echelon of my capture.”

Who's Right?

Both of them

Alexis Murphy - SRE at Jurassic Park

- Primarily protecting PII data for minimum period of time
- GRC does not give explicit crypto requirements
- Adversaries primarily focused on financial crimes and extortion

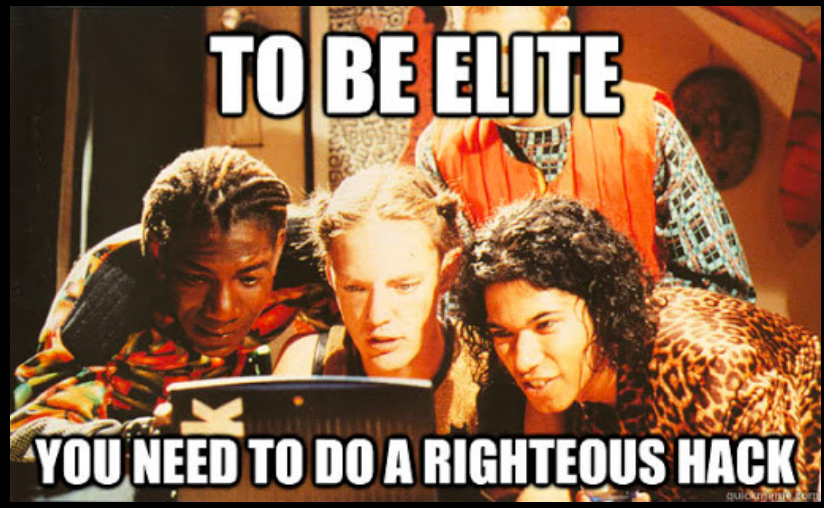
"John" (won't give real name) - SRE at GovCloud

- Protecting US federal data, including data for USMIL and IC
- Secrets protected and archived for long time periods (10yr+)
- GRC gives explicit crypto requirements
- Advanced adversaries including nation stage espionage

To Build Defensively, Think Offensively

"[i]n order to talk about the security of a protocol we need to define the adversarial setting that determines the capabilities and possible actions of the attacker"

D. Dolev, and A. C. Yao, "On the Security of Public Key Protocols," IEEE Transactions on Information Theory, vol. 29, no. 2, pp. 198–208, 1983.



Adversary Modelling

How we think about designing platform and feature security in Vault

When considering features with security implications, we consider the user's potential adversaries: their assumptions, their capabilities, and their goals.

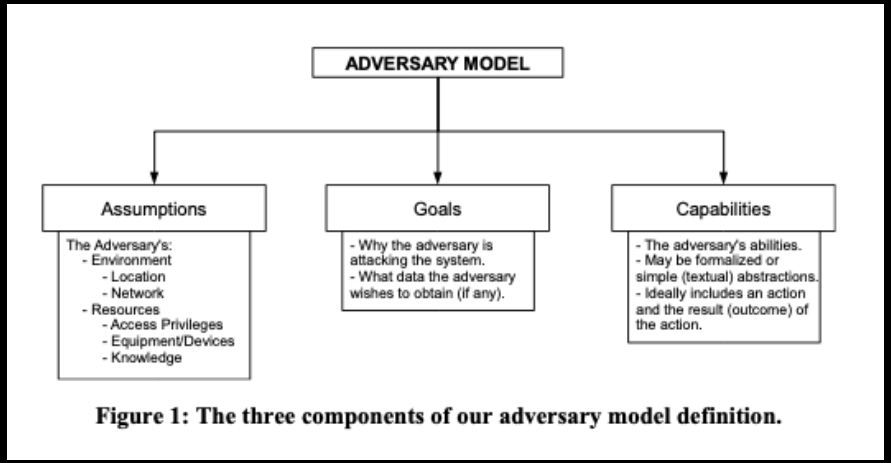


Figure 1: The three components of our adversary model definition.

See: Do, Quang & Martini, Ben & Choo, Kim-Kwang Raymond. (2018). The Role of the Adversary Model in Applied Security Research. Computers & Security. 81. 10.1016/j.cose.2018.12.002.

What is an Adversary?

Who are we defending against?

In Vault, we define an adversary as **anyone who is attempting to access a secret or control who has not been granted explicit access**

Implications:

- Adversaries are not always malicious
- Usability and security go hand in hand

Example of an Accidental Adversary: Harry Potter

Adversary Assumptions

What do we think an adversary brings to the table?

Assumptions can be defined as an adversary's environment and their resources

Environment:
Where is an adversary's location relative to the network / system Vault is running on?

Resources:
What kind of pre-existing resources (privileges or credentials, knowledge of a system, systems for attack or codebreaking) do we assume the adversary possesses?

Sample Adversary Assumptions

Ellington Oil vs. Acid Burn

Environment:
Acid Burn is attacking Ellington systems externally, though she may install local agents through physical infiltration

Resources:
May have stolen privileged credentials via phishing or dumpster diving.

No native knowledge of a system's infrastructure.

No dedicated cryptanalytical HPC infrastructure.

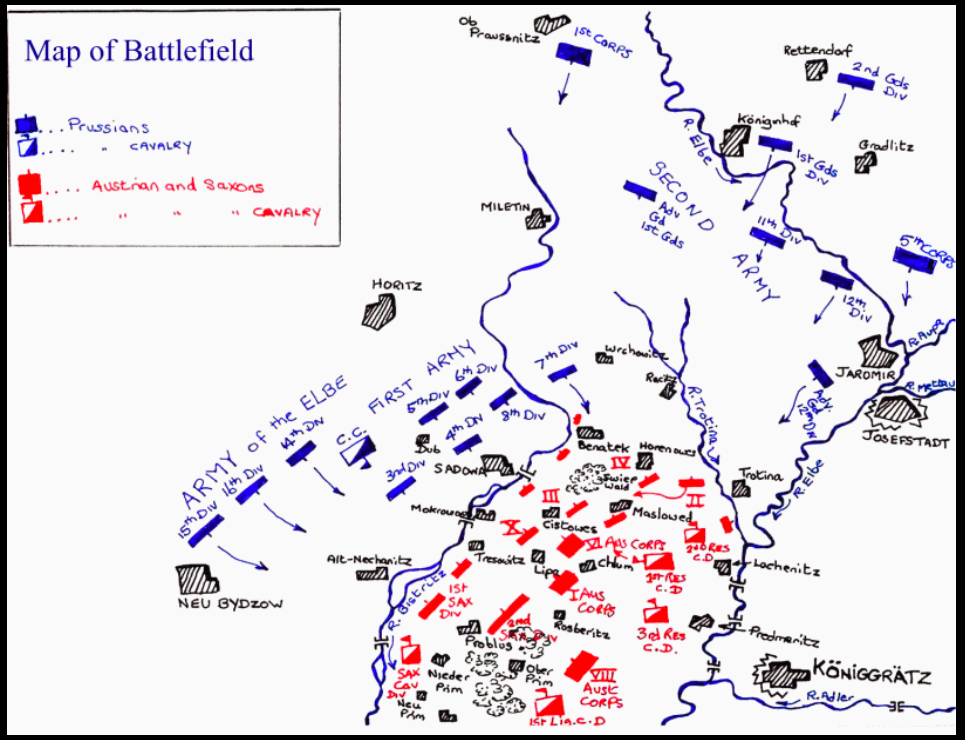
Adversary Goals

Why do we think an adversary is attacking Vault?

Why is an adversary breaching Vault?

What types of data are they attempting to exfiltrate?

Goals are important for identifying what critical security parameters (CSPs) and subsystems we need to build within Vault



Sample Adversary Assumptions

Ellington Oil vs. Acid Burn

Goals:
Attempting to cause chaos in order to cover an attack to exfiltrate data

Likely will be a “shotgun blast” attack to distract defensive resources away from the real attack

Data Targeted:
Anything and everything that will likely be picked up by IDS/IPS and create tickets or alerts

Adversary Capabilities

What can an adversary do to Vault?

Given their assumptions and goals, what kind of attacks is an adversary likely to launch on Vault?

Critical implications on CSPs, how controls are implemented, and acceptance criteria for failure/success of requests

Sample Adversary Capabilities

Ellington Oil vs. Acid Burn

Acid Burn may cause actions to purposely threaten system stability using stale, compromised credentials.

Acid Burn may attempt to aid in the exfiltration of data by costly file searches, thereby causing “noisy neighbor” problems.

Acid Burn may deploy tool-based malware for aggressively breaching systems of compromising stability.

Acid Burn may employ cryptanalysis without dedicated HW.

Using Adversary Models

Designing Vault to withstand attacks from Acid Burn and crew

Given Acid Burn's threat model, Vault must deploy the following to protect itself against adversaries like her:

- Deploy *Resource Quotas* to ensure noisy neighbor activity does not jeopardize system stability.

- Allow for short token TTLs to force Acid Burn to reauthenticate, thereby denying her legitimate access or a means to privilege escalate her allies' attempts to exfiltrate data.

- Ensure the cryptographic barrier is resolved against attacks from high end consumer grade HW.

```
$ tail -f /var/log/vault-audit.log | jq

...snip...
  "response": {
    "mount_type": "token"
  },
  "error": "1 error occurred:\n\t* request path \"auth/token/create\": lease count q
uota exceeded\n\n"
}
```

"Great. There goes MIT."

A World of Adversaries

One platform to defend against many types of adversaries

Just as there's a whole world of different perspectives on security given a user and their use case, there are a whole world of adversaries.

Our goal in Vault is to design features that allow you to "dial up" or "dial down" security features to fit your thread model and the adversaries you are likely to face.

Cryptographic Barrier Architecture

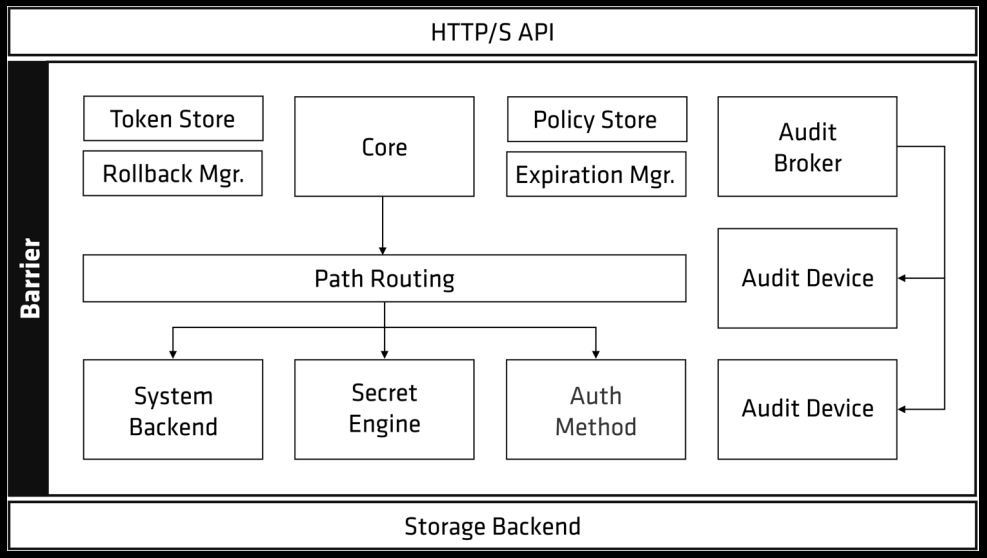
How adversary modeling contributes to feature design

Core Security

Deter an adversary attempting to access Vault's stored data at rest. Ensure that adversary cannot steal key material and it is relatively easy to decrypt Vault storage (*Auto Unseal*).

Protects Against:

Most malicious cybercriminals, hacktivists, non-espionage or codebreaking-capable adversaries.



Cryptographic Barrier Architecture

How adversary modeling contributes to feature design

Advanced Security

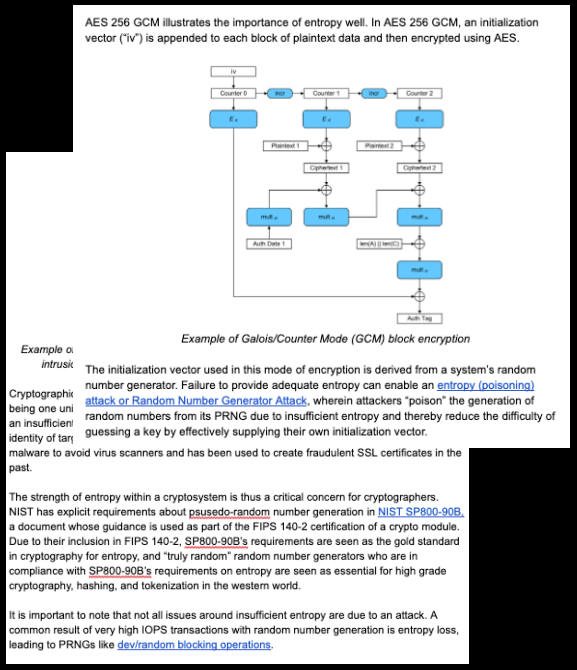
Deter adversaries with dedicated cryptanalytical capabilities by supplementing Vault's crypto barrier with external entropy sources (*Entropy Augmentation*), special purpose hardware (*PKCS#11*), or cryptography from external crypto modules (*Seal Wrap*).

Protects Against:

State-sponsored and nation state adversaries with cryptanalytical capabilities, advanced cybercriminals.

PRDs

Product Requirement Documents



AES 256 GCM illustrates the importance of entropy well. In AES 256 GCM, an initialization vector ("iv") is appended to each block of plaintext data and then encrypted using AES.

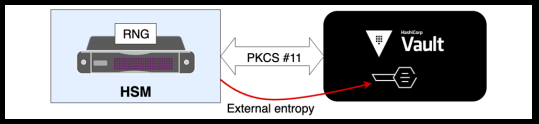
Example of Galois/Counter Mode (GCM) block encryption

The initialization vector used in this mode of encryption is derived from a system's random number generator. Failure to provide adequate entropy can enable an entropy (poisoning) attack or Random Number Generator Attack, wherein attackers "poison" the generation of random numbers from its PRNG due to insufficient entropy and thereby reduce the difficulty of guessing a key by effectively supplying their own initialization vector.

The strength of entropy within a cryptosystem is thus a critical concern for cryptographers. NIST has explicit requirements about psusedo-random number generation in NIST SP800-90B, a document whose guidance is used as part of the FIPS 140-2 certification of a crypto module. Due to their inclusion in FIPS 140-2, SP800-90B's requirements are seen as the gold standard in cryptography for entropy, and "truly random" random number generators who are in compliance with SP800-90B's requirements on entropy are seen as essential for high grade cryptography, hashing, and tokenization in the western world.

It is important to note that not all issues around insufficient entropy are due to an attack. A common result of very high IOPS transactions with random number generation is entropy loss, leading to PRNGs like dev/random blocking operations.

RFCs and Implementation



Adversary Modeling: Kill Your Darlings

Don't be afraid of your adversaries.

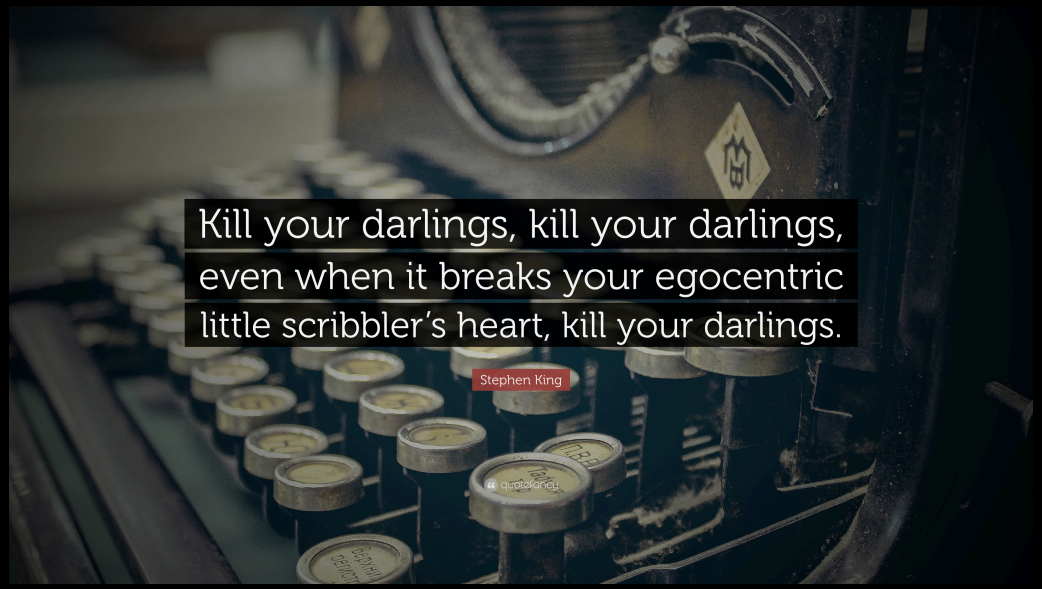
Study them.

Learn from them.

Design for them.

Kill your darlings.

Before your adversaries do.




HashiCorp