Stamp Out Hash Corruption, Crack All the Things!

Ryan Reynolds

- Manager, Crowe Horwath
- Pentester
- Twitter: @reynoldsrb

Jonathan Claudius

- SpiderLabs Security Researcher, Trustwave
- Vulnerability Research
- Twitter: @claudijd

What's inside?

- Windows Hash Extraction
- Story of What We Found
- Windows Hash Extraction Mechanics
- A Different Approach
- Why Are All the Tools Broken?
- Demo
- Patches

LET'S TALK ABOUT HASHES!!!

Goals of Getting Hashes

- Privilege Escalation
- Password Analysis
- Forensics Investigations

Windows Password Hashes

- Two Types of Hashes:
  - LM (Lan Manager)
    - Old Hashing Algorithm w/ Security Flaws
    - Case insensitivity, Broken into 2 Components
  - NTLM (NT Lan Manager)
    - Newer Hashing Algorithm w/ Security Flaws
    - Not salted, but is case sensitive

Windows Password Hashes

- Two Methods to Get Hashes:
  - Injection via LSASS
    - Reads hashes from memory
  - Registry Reading via SAM/SYSTEM
    - Reads hashes from local registry hives

STORY TIME…

Failed Attempt 1

- Social Engineering Engagement
  - Gained Physical Access
  - Dumped Hashes on a Bank Workstation
- Failed to Crack
  - John the Ripper
  - Rainbow Tables

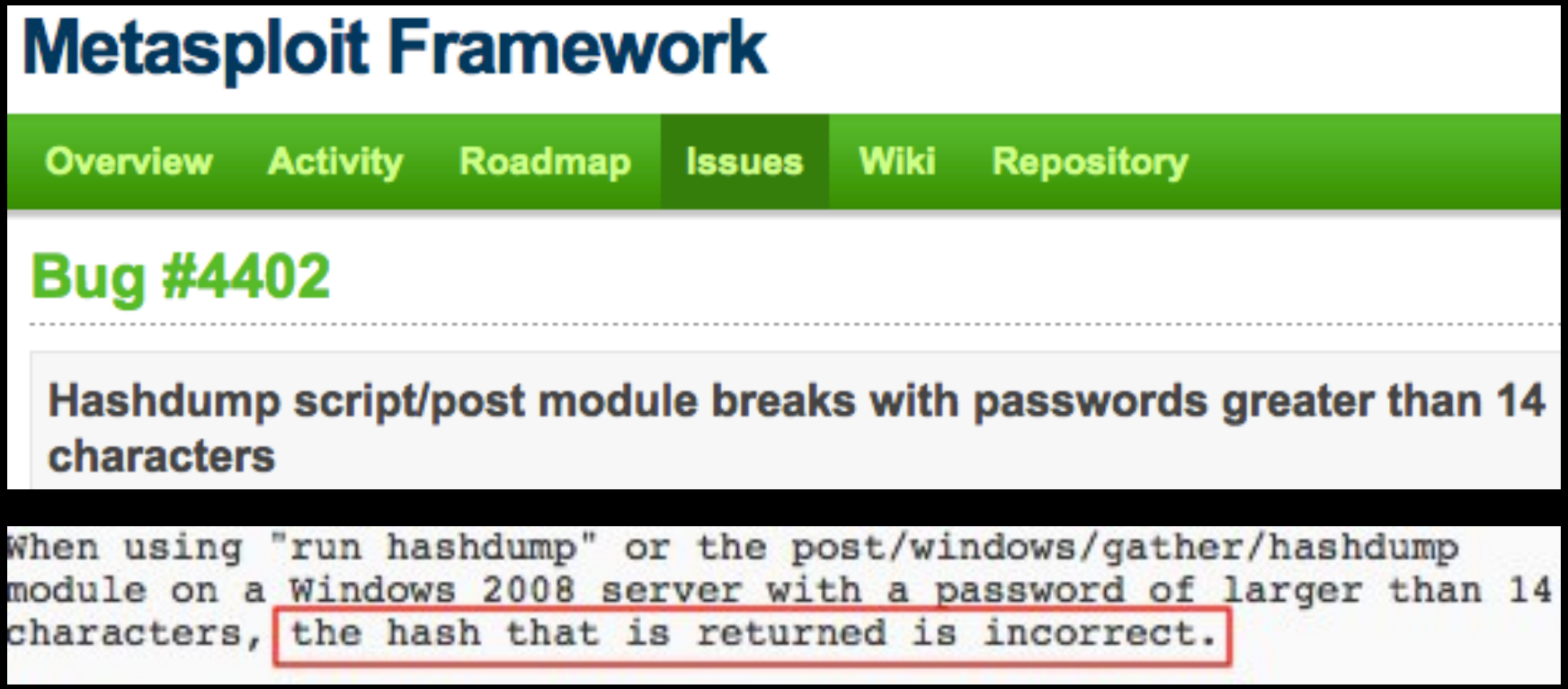
Failed Attempt 2

- Internal Penetration Assessment
  - Popped a Shell via Missing Patch
  - Dumped Hashes on System
- Fail to Crack
  - Rainbow Tables (via all LM Space & French)
  - Pass the Hash (PTH)

Example Hashes

- Via Registry (Metasploit)
  - LM: 4500a2115ce8e23a99303f760ba6cc96
  - NTLM: 5c0bd165cea577e98fa92308f996cf45
- Via Injection (PwDump6)
  - LM: aad3b435b51404eeaad3b435b51404ee
  - NTLM: 5f1bec25dd42d41183d0f450bf9b1d6b

Bug Report


Metasploit Framework
Overview
Activity
Roadmap
Issues
Wiki
Repository
Bug #4402
Hashdump script/post module breaks with passwords greater than 14 characters
When using "run hashdump" or the post/windows/gather/hashdump module on a Windows 2008 server with a password of larger than 14 characters, the hash that is returned is incorrect.

“Our Powers Combined…”

- Beers
- Hacking
- More Beers

Example Hashes

- Via Registry (Metasploit)
  - LM: 4500a2115ce8e23a99303f760ba6cc96
  - NTLM: 5c0bd165cea577e98fa92308f996cf45
- Via Injection (PwDump6)
  - LM: aad3b435b51404eeaad3b435b51404ee
  - NTLM: 5f1bec25dd42d41183d0f450bf9b1d6b

WHERE DO HASHES LIVE?

Where Do Hashes Live?

- HKLM\SAM
  - Store security information for each user (including hash data)
- HKLM\SYSTEM
  - Stores the SYSKEY ("salts" the SAM information for security purposes)

What The Registry Looks Like

- HKLM\SAM\SAM\domains\account\users\
  - Users: 000001F4, ..1F5, etc.

Name	Type	Data
(Default)	REG_SZ	(value not set)
F	REG_BINARY	02 00 01 00 00 00 00 00 8d
V	REG_BINARY	00 00 00 00 bc 00 00 00 02

What's Inside These Values?

- For each user, we have two values…
  - "F" – Binary Data
    - Last Logon, Account Expires, Password Expiry, etc.
  - "V"- Binary Data
    - Username, LM Hash Data, NT Hash Data, etc.

A Closer Look At Raw Data

- Raw Data w/ LM & NTLM Data

```
...0000AAAAAAAA0000BBBBBBBB00000...
```

- Raw Data w/ just NTLM Hash Data

```
...00000000BBBBBBBB000000000000...
```

Registry Extraction Tools

- Metasploit Hashdump Script
- Creddump
- Samdump2
- Cain and Able
- Pwdump7
- FGDump 3.0
- Others

Current Parsing Logic

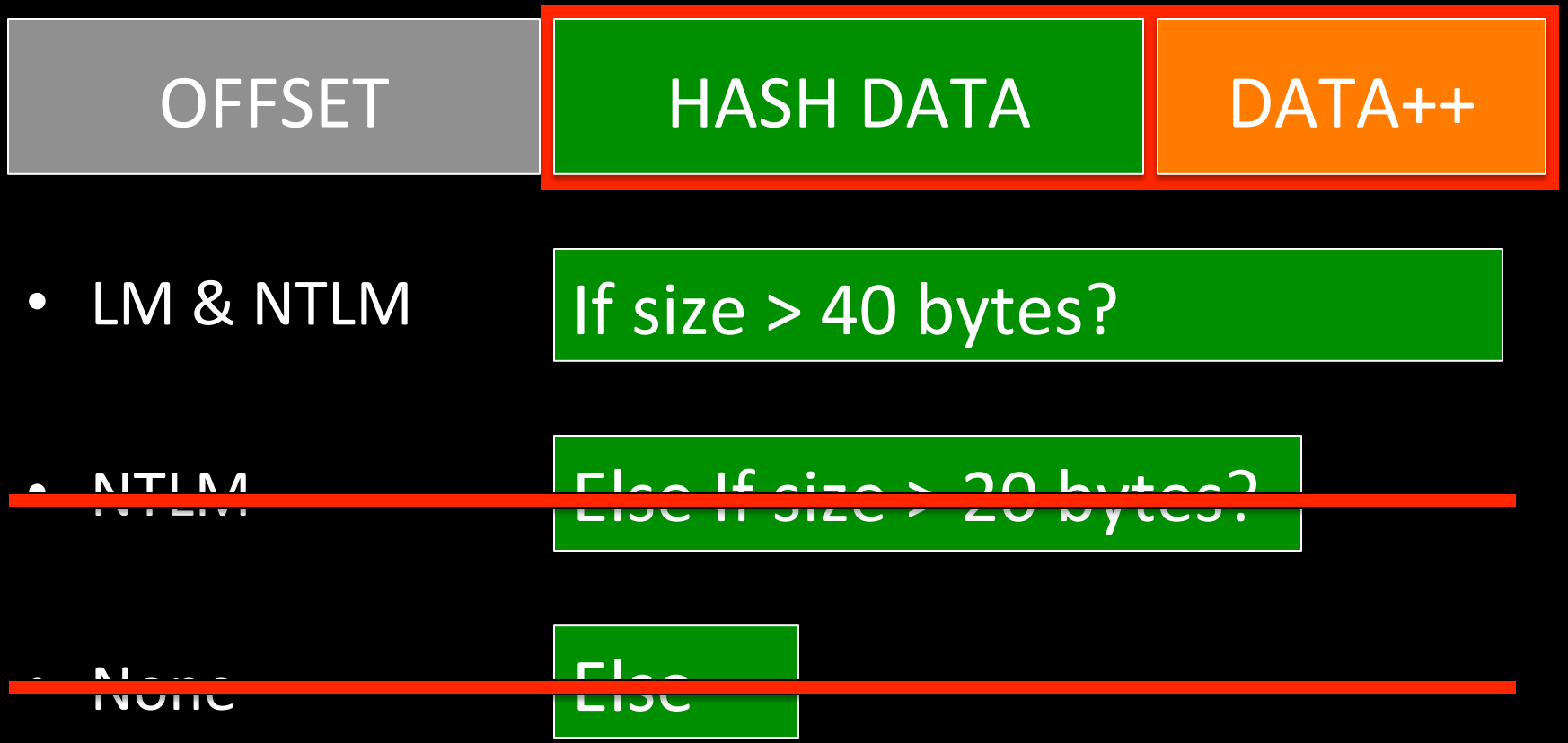
OFFSET	HASH DATA
• LM & NTLM	If size > 40 bytes?
• NTLM	Else If size > 20 bytes?
• None	Else

THE “FLAW”

Remember These?

- Via Registry (Metasploit)
  - LM: 4500a2115ce8e23a99303f760ba6cc96
  - NTLM: 5c0bd165cea577e98fa92308f996cf45
- Via Injection (PwDump6)
  - LM: aad3b435b51404eeaad3b435b51404ee
  - NTLM: 5f1bec25dd42d41183d0f450bf9b1d6b

The “Flaw”



The “Flaw”

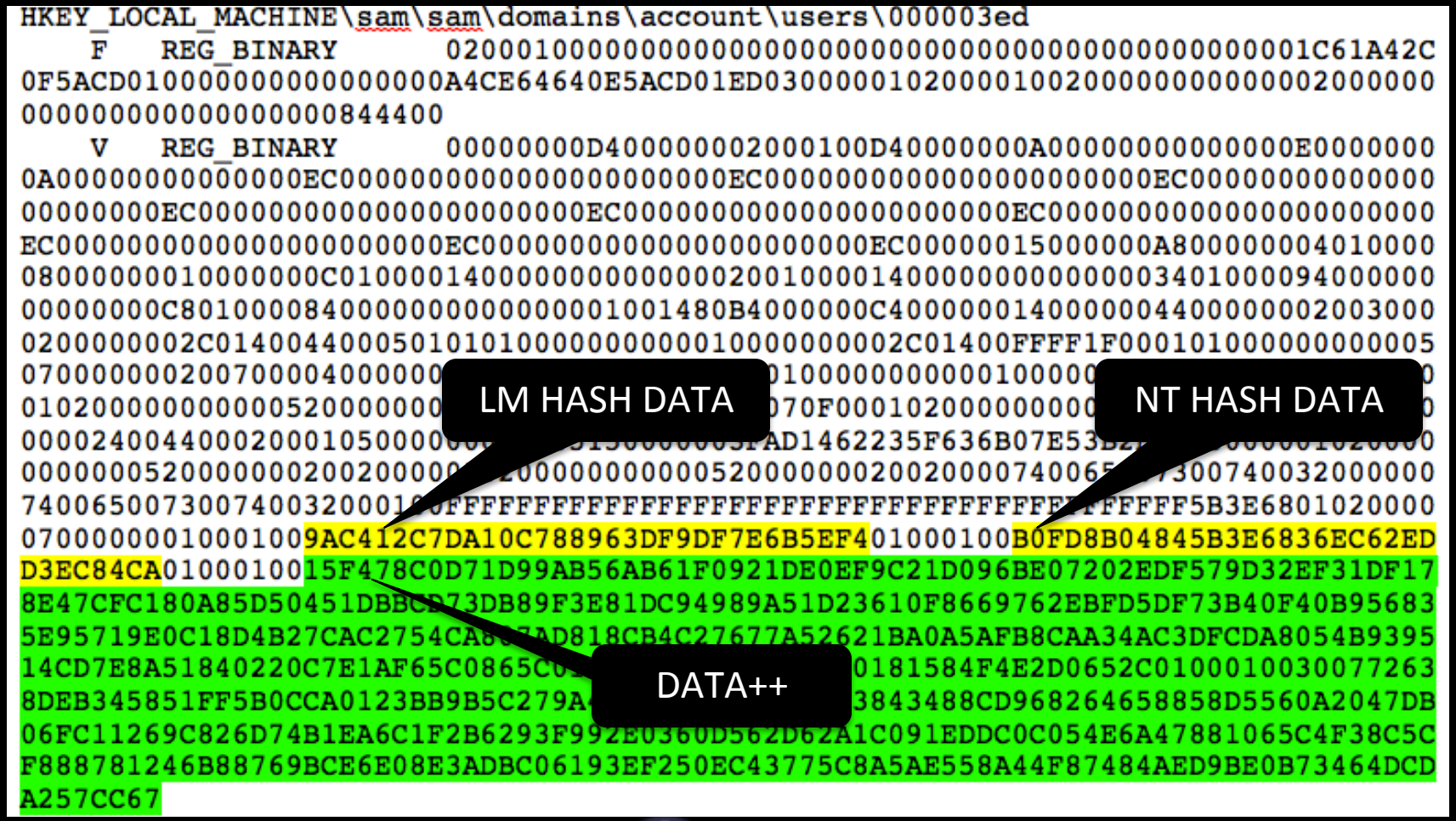
BAD

Root Cause?

- How do we get “DATA++”?

- By following Microsoft best practices
  - Set Password History
  - No LM Hashes

Raw Look at “V” Data Structure



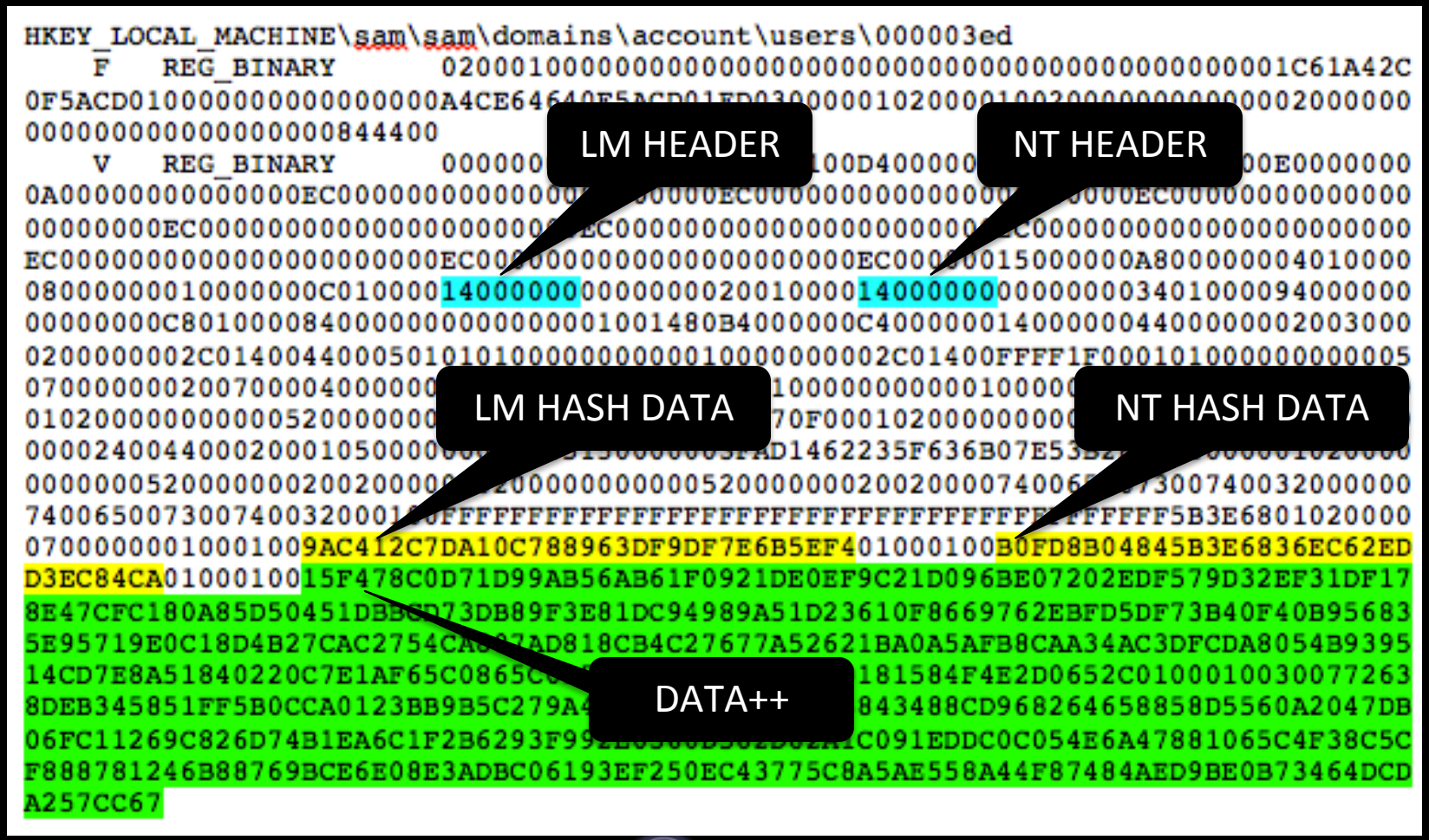
How often does this occur?

- Newer OS's do not store LM
  - Windows Vista and newer
  - LM can be disabled by a proactive Sysadmin
- Password histories set through GPO

In an ideal world…

- We would want…
  - LM Exists?
  - NTLM Exists?
  - Parse correct hash data 100% of the time

Raw Look at “V” Data Structure



A Different Approach

- “V” hash 4 byte headers for LM & NTLN
  - 0x4 (4 bytes) = Hash Not Present (false)
  - 0x14 (20 bytes) = Hash Present (true)

- No more guessing!

A Different Approach

OFFSET	HASH DATA	DATA++
• LM & NTLM	If LM.exists? && NTLM.exists?	
• NTLM	Else If NTLM.exists?	
• None	Else	

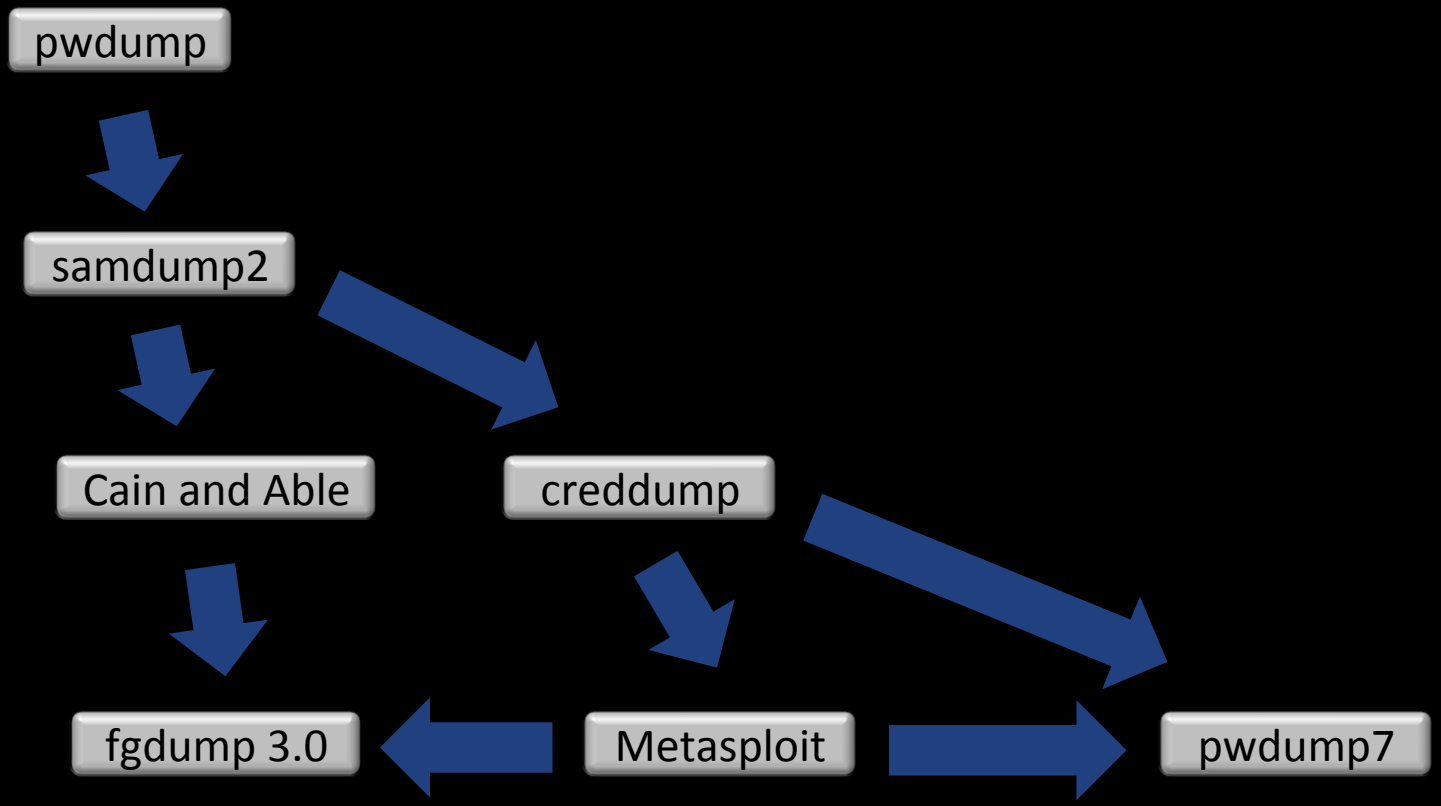
A Different Approach

BAD LOGIC

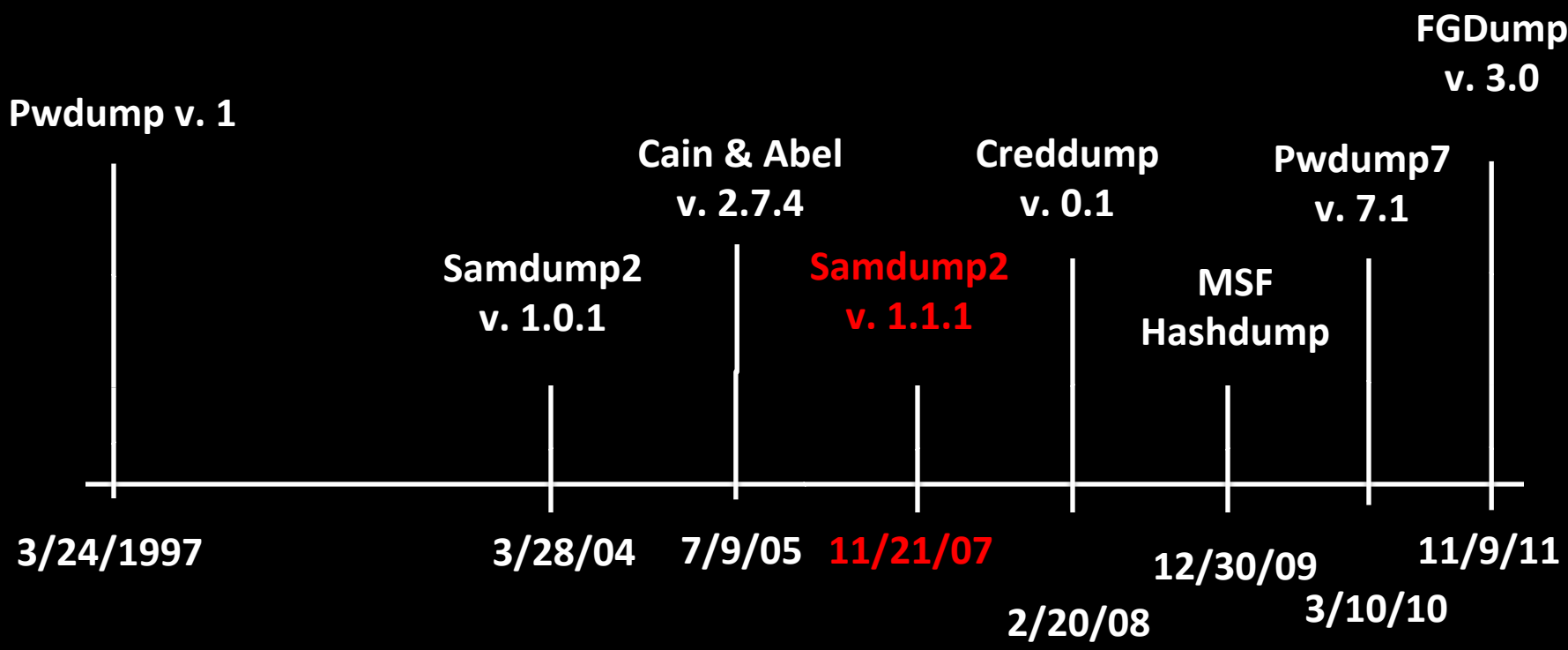
GOOD LOGIC

WHY ARE ALL THE TOOLS BROKEN?

Who's Patient Zero?



Tool Timeline



Take Away

- Reverse engineering is hard
  - Exhaustive testing is time consuming
- Leveraging code is helpful
  - Fully reusing code is not always good
- Open source let's others learn and help fix!

DEMONSTRATION

PATCHES!!!

Patches

Affected Tools	Patched?
Creddump	Yes
Metasploit's Hashdump Script	Yes
L0phtcrack	Working with Author(s)
Pwdump7	Working with Author(s)
FGDump 3.0	Working with Author(s)
Samdump2	Fixed in v 1.1.1
Cain & Abel	Working with Author(s)

QUESTIONS?