Screen Range Test
Cryptohaze Design & Architecture

Bitweasil
cryptohaze.com
How did this start?

Grad Student + Rainbow Tables +

= Thesis!
Multiforcer History

- 2008: C/CUDA for rainbow table prefilter
- 2010: C++/CUDA rewrite
- 2011: Initial network support
- 2012: C++/CUDA/OpenCL/CPU rewrite
  - Goal: No Duplicate Code (or very little)
Technologies - 2007

SSE/SSE2
Technologies - 2012

SSE\{2,3,4\}
XOP/AVX

Tuesday, December 4, 12
Cryptohaze Multiforcer

- GPLv2
- Linux, Windows, OS X
- CUDA, OpenCL, CPU
- C++ Framework
- Integrated Network Support
Code Architecture

- **{Tech}_{Hash}.{cl,cu}**: Device kernel, making use of common code libraries/building blocks
- **Plain{Tech}_{Hash}**: Kernel specific setup and transfer
- **Plain{Tech}**: Compilation, memory management, main data copying to device
- **HashTypePlain**: Logic, control flow, main operations
Code Architecture

- CUDA_MD5.cu
- CUDA_NTLM.cu
- CUDA_IPB.cu

PlainCUDA_MD5
PlainCUDA_NTLM
PlainCUDA_IPB

PlainCUDA
PlainCPU
PlainOpenCL

HashTablePlain

Tuesday, December 4, 12
Code Architecture

OpenCL_MD5.cl  OpenCL_NTLM.cl  OpenCL_IPB.cl

PlainOpenCL_MD5  PlainOpenCL_NTLM  PlainOpenCL_IPB

PlainCUDA  PlainCPU  PlainOpenCL

HashTypePlain
Workunits

- Workunit ID
- Start Point
- End Point
- Password Length
- Flags
- Wordlists
- Client ID
# Workunits

<table>
<thead>
<tr>
<th>Workunit ID</th>
<th>Start Point</th>
<th>End Point</th>
<th>Password Length</th>
<th>Flags</th>
<th>Wordlists</th>
<th>Client ID</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Generation Parameters</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Pending Queue</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Inflight Queue</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Done</th>
</tr>
</thead>
</table>
Import/Exports

```cpp
void ImportHashListFromRemoteSystem(
   std::string &remoteData);

void ExportHashListToRemoteSystem(
   std::string &exportData);
```
Serialization: Protocol Buffers

- Efficient, binary-packed, machine-parseable format (vs XML)
- Backwards compatible with optional fields
- Each class only talks to another instance across the network
- Network class just passes a stream of data
The Problem with Wordlist Support

- Varying maximum length supported
- Single file source, or single stdin support
- Network distributed support is hard
Cryptohaze Wordlists

- Wordlists
  - JTR
  - Hashcat CPU
  - Custom Scripts
  - Markov/Sentences

- Cracking Server
  - Port 4444
  - Local Devices
  - Remote Clients

Supports length 0-127
MD5, NTLM, IPB
WebTables
“Tables in the Cloud”

deadbeef
d00df00d

Candidate Hashes

p@s5w0rd

Chains to Regen

Tuesday, December 4, 12
Candidate Hash Generation

Reduce 0
- 'passw0rd'
- Hash
  - bed12836...
- Hash
  - 1a2b3c4d...

Reduce 1
- 'secret!!'
- Hash
  - ebd86c21...
- Hash
  - 5bba1e61...

Reduce 2
- '1q2w3e4r'
- Hash
  - 5416d7cd...
- Hash
  - 25d55ad2...

Hash
- 86f445c3...
- 1a2b3c4d...
Candidate Hash Generation

Reduce
‘passw0rd’
Hash
bed12836...
Reduce
‘secret!!’
Hash
ebd86c21...
Reduce
‘1q2w3e4r’
Hash
5416d7cd...

‘2private’
5bbale61...
‘12345678’
86f445c3...

‘aoeuhtns’
25d55ad2...
‘aoeuhtns’
86f445c3...

Reduce
1a2b3c4d...
Hash
Reduce
1a2b3c4d...
Hash
Reduce
1a2b3c4d...
Hash
Chain Search

1a2b3c4d... 25d55ad2... 5416d7cd... 86f445c3...

WebTables Server

‘fr&hb%hi’ '/gxE}{a(5’ ‘){aXzYXe’ ‘R)mq%YOj’
WebTables Search

24 bits

HTTP POST

1a2b3c25d55a5416d786f445
WebTables Server

1a2b3c25d55a5416d786f445

Table Server

Table Server

Table Server

Table Server

fr&hb%hi\n/gxE}a(5\n!}aXzYXe\nR)mq%YOj

Tuesday, December 4, 12
WebTables

- Last 2 candidate chains skipped normally, user-selectable count
- 0.001% reduction in efficiency (with 200k chain length)
- Flexible backend - transparent to end user
- Zero table download!
Questions?

- cryptohaze.com
- webtables.cryptohaze.com
- bitweasil@cryptohaze.com
- #cryptohaze on irc.freenode.net