QUICK INTRODUCTION

Currently: Senior Consultant at CrowdStrike

Previously: Large-scale detection at Fortune 5

Bro user for 2+ years

Focus on network forensics and incident response

Twitter: @jsh1brd
GOALS FOR THIS TALK

You'll learn something new about RDP

You'll see one of the newest Bro analyzers in action

You'll leave with some useful methods to find bad guys in your network
WHAT'S THE DEAL WITH RDP?
RDP KEY POINTS

Enables remote system access across the network

Connection is encrypted

Definitely being used in your organization
WHY I'M TALKING ABOUT RDP

Bro 2.4 has an RDP analyzer!
WHY THIS ANALYZER EXISTS
Everything that happens over TCP ->

We care about a **very** small part of this
- Connection Initiation
- Basic Settings Exchange
PROTOCOL DETAILS
X.224 CONNECTION REQUEST (C)

Client initiates connection
- Client-supported security protocols
- Connection correlation identifier
- Optional routing token / cookie
PROTOCOL DETAILS
X.224 CONNECTION CONFIRM (S)

Server responds to connection initiation
- Successful? Server selected protocol
- Unsuccessful? Reason request failed
Client sends settings data
- Client computer name
- Keyboard language settings
- RDP client version
PROTOCOL DETAILS
MCS CONNECT RESPONSE (S)

Server sends response settings data
- RDP server version
- Encryption method and level
- Server certificate
PROTOCOL
CHALLENGES
PROTOCOL CHALLENGES

ENCRIPTION!

No cookie == no identifiable packet data
Most forensically useful metadata is optional
- Cookie
- Client computer name
PROTOCOL CHALLENGES
COOKIE!

Length ranges from 9 to ~127 characters

Introduces 'user collision'
- Multiple users appear to be one user

15 chars: DOMAIN\samantha
09 chars: DOMAIN\sa
12 chars: DOMAIN\sally
09 chars: DOMAIN\sa
IDENTIFYING RDP
IDENTIFYING RDP IN THE RAW

   ...$.....Cookie: mstshash=A70067..
#
   ...
#
   ........4.
#
   ........e..........0.`.."0............................0.....................
   .............
   ..&...Duca..`
   ........................................(..I.S.D.2.-K.M.8.4.1.7.8..............
   .............................................................5.5.2.7.4.-.0.E.
   M.-0.0.1.1.9.0.3.-.0.0.1.0.7........................................rdpsnd.....clip
   rdr.....rdpsnd.....
#
   ...M..f.A......0.`".................................|..v..McDn......................
   ......................................w......=6.....R.r0....b."f.3r................\RSA1H..........?................|..Zr...
   \..F.p.:X..........k&.b..8[Z..._])...C................................H..rI.x/./}L.../1d.`......h=.g....#
   u.vz........G.. .NT.oja..W.%..?........
IDENTIFYING RDP
DETECTION STRINGS

...$.......Cookie: mstshash=A70067..
#
.....
#
.........4.
#
.............e........0...."...................0.........................0............................
........../...|.....&.......Duca................``........(...I.S.D.2...K.M.8.4.1.7.8..............
............................................................5.5.2.7.4...0.E.
M.-0.0.0.1.0.9.0.3.-0.0.1.0.7....................................................rdpdr..clip
rdr.....rdpsnd......
#
...M...f..A..0..."..................................|...v.....McDn......................
..........................w....=6....R.r0....b.."f.3r.................\RSA1H......?............|..Zr..
\....F.p...X.........k&.b..8[Z..__)...C..........................H...rI.x../}L.../1d.`....h=..g....#
u.vz........G...NT.oj)..W..%..?........
IDENTIFYING RDP DETECTION STRINGS++

```
# Cookie: mstsshash=

# T 10.226.29.74:3389 -> 10.226.41.226:13178 [A]
.....
# T 10.226.29.74:3389 -> 10.226.41.226:13178 [AP]
.....4.
# T 10.226.41.226:13178 -> 10.226.29.74:3389 [AP]
.....e.0....................0........................................0.................................
..........................................................Duca.............................(.I.S.D.2.-K.M.8.4.1.7.8........................
..........................................................5.5.2.7.4.-0.E.
M.-0.0.1.1.9.0.3.-0.0.1.0.7..................................................,rdpdr........clip
 rdr....rdpsnd......
# T 10.226.29.74:3389 -> 10.226.41.226:13178 [AP]
.....f.A.0..................\..................McDn.................................w......=6....R.r0....b."f.3r........................\RSA1H......?................|..Zr..\F.p:..X..........k&.b..8[Z...].C..................H...rI.x/.}L.../1d.`......h=..g....#
u.vz........G...NT.oja..W%.?........
```
IDENTIFYING RDP DETECTION STRINGS++

```
# Cookie: mstshash=
...........
#
.........4.
#
............0......................0..........................
.............\|..&..Duca...........
...............\I.S.D.2.-K.M.8.4.1.7.8.....................
...............\5.5.2.7.4.-0.E.
M,-0.0.1.1.9.0.3,-0.0.1.0.7..........................clip
rdr.....rdpsnd......
#
.........M...f.A........0.......|..*..v........McDn..........................
.............w.......=6.....R.r0.....b.."f.3r.........\\.RSA1H........?
...........\F.p.:X...........k&.b.8[Z__]...C..............H..rI.x/.}L.........id.```

u.vz........G.. .NT.oja..W%..?........
```
IDENTIFYING RDP DETECTION STRINGS++

Cookie: mstsshash=

......

......4.

......

......

rdspsnd

......

......
IDENTIFYING RDP DETECTION STRINGS++

Cookie: mstsshash=

```

#

Cookie: mstsshash=


.......

#


.......

#


```

rdpsnd

```

0.0.1.0.7.................................

```

rdpdr

```

McDn............................

```

```
event connection_state_remove(c: connection)
{
    if ( c$id$resp_p == 3389/tcp
        && c$conn$orig_bytes >= 1000
        && c$conn$resp_bytes >= 1000 )
    print "found RDP?";
}
IDENTIFYING RDP
<= BRO 2.3++

signature dpd_rdp_client {
    ip-proto == tcp
    # Client request
    payload /.*(Cookie: mstshash\=|Duca.*(rdpdr|rdpsnd|drdynvc|cliprdr))/
    requires-reverse-signature dpd_rdp_server
    enable "rdp"
}

signature dpd_rdp_server {
    ip-proto == tcp
    payload /./(5)\xd0|.McDn)/
}

(Actually the dpd.sig for RDP in Bro 2.4)
IDENTIFYING RDP
THE PROBLEM (UNTIL NOW)

Network detection isn't useful
Network detection doesn't scale
Detecting RDP on the network wastes analyst time
IDENTIFYING RDP
BRO 2.4

cookie: A70067
keyboard_layout: English - United States
client_build: RDP 5.1
client_hostname: ISD2-KM84178
desktop_width: 1152
desktop_height: 864
result: Success
security_protocol: RDP
encryption_level: High
encryption_method: 128bit
IDENTIFYING RDP ANALYZER CAVEATS

It's not magic
- Won't identify RDP over SSL
- Won't identify RDP over SSH

It's most useful when monitoring internal-to-internal sites

"Success" != successful authentication
- Still need to validate with non-network data
RDP HUNTING
RDP HUNTING
A QUICK NOTE ON HUNTING ...

Hunting is a proactive approach to identifying threats on the network.

It gives you the opportunity to identify new types or new variants of threats.

Many things affect your ability to hunt:
- Knowledge
- Skillset
- Toolset
- Leadership
RDP HUNTING
A QUicker NOte ON RDP MetADAtA

You have to hunt through it
- IOCs (IP addresses) won't help you
- IDS alerts will waste your time
RDP HUNTING
BRO HUNTING METHODS

Stacking
- Simple outlier analysis
- Complex outlier analysis

Tracking
- Using inside knowledge to identify attacker activity

Timelines
- Monitoring activity across a distinct range of time
RDP HUNTING
SIMPLE STACKING

Primary use: identify new users and computers in the network

Identify new users in the network
bro-cut cookie < rdp.log | sort | uniq -c | sort -n

Identify new computers in the network
bro-cut client_name < rdp.log | sort | uniq -c | sort -n
RDP HUNTING
COMPLEX STACKING

Primary use: identify scanning and worms, compromised user accounts

Identify users connecting to a high number of systems

sourcetype=bro source=*rdp* cookie=* | stats dc(dest_ip) AS dc_dest_ip by cookie
RDP HUNTING
COMPLEX STACKING++

Identify multiple users on a single computer

sourcetype=bro source=*rdp* client_name=* cookie=* | stats values(cookie) dc(cookie) AS dc_cookie by client_name | where dc_cookie > 1
RDP HUNTING TRACKING

Primary use: identify lateral movement

Dependencies
- Knowledge of network and organization
- Accessible, organized data
RDP HUNTING TRACKING++

Scenario
- Sensor A monitors traffic between business units X and Y
- Net block B belongs to business unit X
- Net block C belongs to business unit Y
- RDP between the two is uncommon
- Business unit Y develops high-value projects
Identify users accessing abnormal sections of the network

`sourcetype=bro source=*rdp* cookie=* sensor=a
  ( tag::src_ip=nb_b tag::dest_ip=nb_c )
OR ( tag::src_ip=nb_c tag::dest_ip=nb_b )
| stats count by src_ip,dest_ip,cookie`
RDP HUNTING TRACKING++

Identify computers accessing abnormal sections of the network

sourcetype=bro source=*rdp client_name=* sensor=a ( tag::src_ip=nb_b tag::dest_ip=nb_c ) OR ( tag::src_ip=nb_c tag::dest_ip=nb_b ) | stats count by src_ip,dest_ip,client_name
RDP HUNTING TIMELINES

Primary use: identify anomalous access

Effective use is dependent on how much data you have
- Search all computers vs. single computer

Identify access time by computer

```
sourcetype=bro source=*rdp* client_name=* 
| timechart useother=F span=1hr count by client_name
```
CASE STUDIES
SCANNING / WORMS

Fairly easy to identify when hunting – they’re noisy

Found by stacking cookie X id.resp_h
- Look for users to connect to a high number of systems

Especially useful if you isolate events into periods of time
- User A connected to N number of systems in T minutes
## CASE STUDIES
### SCANNING / WORMS++

One week of RDP activity

<table>
<thead>
<tr>
<th>-cookie</th>
<th>uniq # id.resp_h</th>
</tr>
</thead>
<tbody>
<tr>
<td>rdp_logon_screen.nbin</td>
<td>1384</td>
</tr>
<tr>
<td>os_fingerprint_rdp.nbin</td>
<td>1375</td>
</tr>
<tr>
<td>Administr</td>
<td>253</td>
</tr>
<tr>
<td>a</td>
<td>30</td>
</tr>
<tr>
<td>a</td>
<td>25</td>
</tr>
</tbody>
</table>

Note: the search from slide 34 can identify this activity
CASE STUDIES
SCANNING / WORMS++

One week of RDP activity

cookie[count]                   threat
dpd_logon_screen.nbin[1384]     Nessus
os_fingerprint_rdp.nbin[1375]   Nessus
Administr[253]                  Collision
[30]                            ???
a[25]                            Morto worm
Identifying inbound attacker access w/ RDP metadata is a difficult game to win.

Monitoring VPN nodes is the best chance to identify remote attackers.

Scenario
- Single factor VPN
- Dealing with potentially compromised user accounts
CASE STUDIES
REMOTE ATTACKER ACCESS++

Identified attacker connecting to the network via VPN

Found by tracking inbound connections between 2:00 and 12:00 UTC

#fields     keyboard_type   keyboard_layout client_build
  client_name     client_dig_product_id   desktop_width       desktop_height

Japanese    English - United States RDP 7.1
<client_name>   <client_dig_product_id >    1576    928
Japanese    English - United States RDP 5.2
<client_name>       (empty)     1576    928
Japanese    English - United States RDP 5.2
<client_name>       (empty)     1576    928
Japanese    English - United States RDP 7.1
<client_name>   <client_dig_product_id >    1576    928
Could’t rely on attacker always connecting from the same VPN node

Could rely on client_name, desktop_width, and desktop_height remaining the same

<table>
<thead>
<tr>
<th>fields</th>
<th>keyboard_type</th>
<th>keyboard_layout</th>
<th>client_build</th>
<th>client_name</th>
<th>client_dig_product_id</th>
<th>desktop_width</th>
<th>desktop_height</th>
</tr>
</thead>
<tbody>
<tr>
<td>Japanese</td>
<td>English - United States RDP 7.1</td>
<td>&lt;client_name&gt;</td>
<td>&lt;client_dig_product_id&gt;</td>
<td>1576 928</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Japanese</td>
<td>English - United States RDP 5.2</td>
<td>&lt;client_name&gt;</td>
<td>(empty)</td>
<td>1576 928</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Japanese</td>
<td>English - United States RDP 5.2</td>
<td>&lt;client_name&gt;</td>
<td>(empty)</td>
<td>1576 928</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Japanese</td>
<td>English - United States RDP 7.1</td>
<td>&lt;client_name&gt;</td>
<td>&lt;client_dig_product_id&gt;</td>
<td>1576 928</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
QUESTIONS?
REFERENCES


» http://www.snakelegs.org/2011/02/06/rdp-cookies-2/