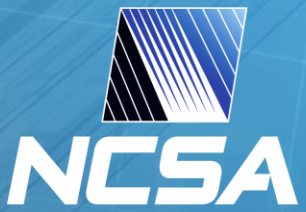
How We Use Bro at NCSA

Sam Oehlert



National Center for Supercomputing Applications
University of Illinois at Urbana–Champaign

Why We Use Bro At NCSA

- Security engineer = operational security
- Incident Response leans heavily on Bro

Bro's Triumphs

- Malware Hash Registry
- Message: 192.168.1.3 md5hash http://badurl.com/dangerousfile

Connection: 192.168.1.3->1.2.3.4

orig/src hostname: host.ncsa.illinois.edu

resp/dst hostname: badhost.com

Bro's Triumphs (II)

- Incorrect File Type
- Message: application/x-dosexec GET http://badhost.com/badfile.txt

  Connection: 192.168.1.3-> 1.2.3.4

  Connection uid: DoRC4Vi2LB6

  orig/src hostname: host.ncsa.uiuc.edu

  resp/dst hostname: badhost.com

Bro's Triumphs (III)

- SQLI
- Message: An SQL injection victim was discovered!

  Address: 192.168.1.3

  SQL Injection samples

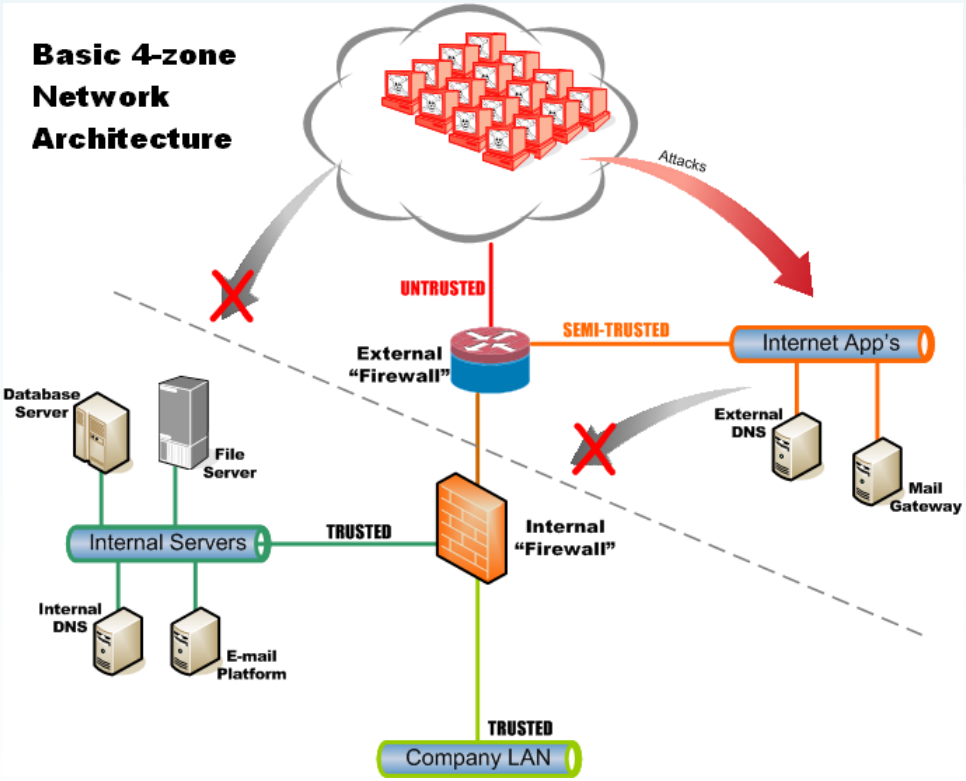
  /index.php?module=subjects&func=viewpage&pageid=1%20UNION%20select%201,1,1,pass,uname,",",",",",",",",0,1,1,1,1%20from%20users%20where%20uname='Admin'

Other Uses for Bro

- Asset management
- Misconfigurations
- Security Issues

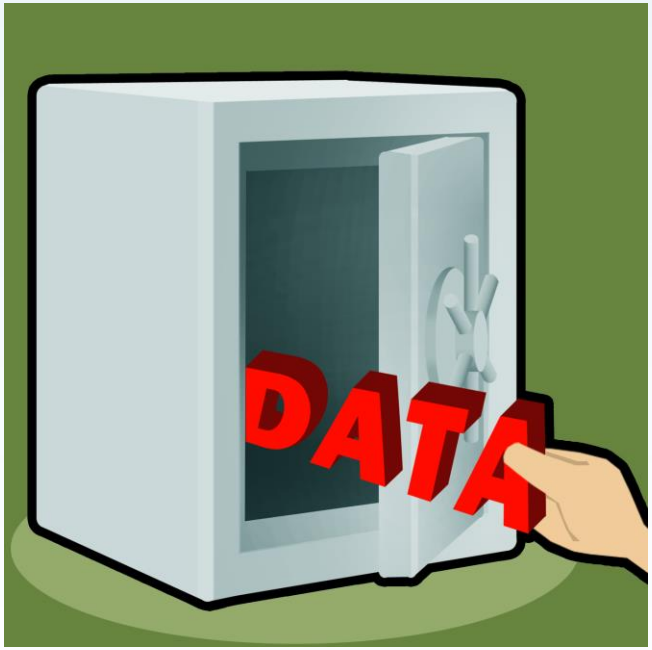
Vetting

- Security Zones
- How can you know your network if you don't know what hosts exist?



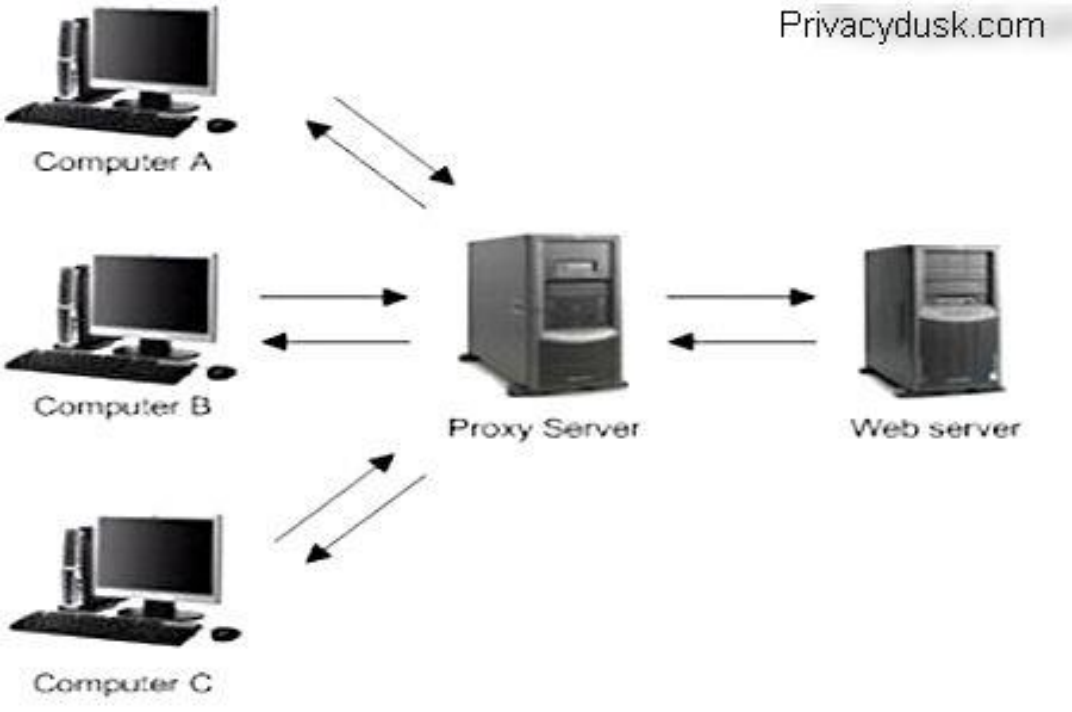
Syslog

- How to get everyone to play how we want?
- Protect sensitive info



Open Proxy Detection

- Are our hosts being too nice?
- Bro finds socks proxies already



Hidden Spam

- Go to website, all looks fine
- Google search, pharma spam abounds

```
const hidden_spam_name =
        /[vV][iI][aA][gG][rR][aA]/                  |
        /[cC][iI][aA][lL][iI][sS]/                  |
        /[pP][hH][aA][rR][mM][aA][cC][yY]/          |
        /[cC][aA][nN][aA][dD][aA]/                  |
        /[cC][aA][nN][aA][dD][iI][aA][nN]/          |
        /[aA][uU][sS][tT][rR][aA][lL][iI][aA]/  &redef;
```

Bro With Auditing SSHD

- Patched SSHD to capture commands/output
- Bro to alert based on data

Flashback Trojan

- No insight to it on our network
- User agent/URI check

```
function check_flashback_ua(c: connection)
        {
        # Checking for "id:<-style string>" in the user_agent
        if (/ id\:[a-fA-F0-9]+\-[a-fA-F0-9]+\-[a-fA-F0-9]+\-[a-fA-F0-9]+\-[a-fA-F0-9]+\)/ in c$http$user_agent)


function check_flashback_uri(c: connection)
        {
        # Now check for the right strings
        if (/^\/(stat_u|stat_n|stat_d|scheck|stat_svc|auupdate|owncheck)\// in c$http$uri)
```

Java Drive By

- Many of our issues are drive by malware
- User education can only do so much

```
event connection_state_remove(c: connection)
        {
        if ( ! Site::is_local_addr(c$id$orig_h) ) return;
        if ( ! c?$http ) return;
        if ( ! c$http?$user_agent ) return;
        if ( bad_user_agents !in c$http$user_agent ) return;
        if ( c$http?$mime_type && c$http$mime_type == "application/x-dosexec" )
                do_java_drive_by_notice(c);
        else if ( c$http?$uri && /\.exe/ in c$http$uri )
                do_java_drive_by_notice(c);
```

Known Bad

- Honeypots are useful
- Hacking becomes patterned

```
const sensitive_URIs =
        /.*0304-exploits\.sormail\.c/        |    ## exploit package
        /.*0x333shadow\.tar\.gz/             |    ## honeypot download
        /.*21book\.zip/                      |    ## feebs.worm
        /.*64sys.*/                          |    ## honeypot download
        /.*6_crondDts\.tgz/                  |    ## honeypot download
        /.*8_go\.tgz/                        |    ## honeypot download
        /.*9xq\/_gate\.php/                  |    ## zeus PII stealing bot
```

Where to Next?

- Sumstats
- Active Response
- Internal Defense