Monitoring Remote Locations with Nagios

John Sellens
jsellens@syonex.com

October 3, 2013

Notes PDF at http://www.monbox.com/notes/
“There’s More Than One Way To Do It”

Notes:

- Not a great idea for a programming language
- But sometimes handy when solving real problems
- I’m not the world’s biggest Perl fan
Monitoring Remote Locations with Nagios

Setting the Scene

- Sometimes everything you care about is on one network
  - Single security zone
  - Acceptable latency and loss

- Sometimes things you care about are all over the place
  - Different locations
  - Different security zones
  - Different owners
  - Unpredictable or unstable links
**What Will We Cover?**

- Remote monitoring situations you may run into
- Techniques for monitoring remote devices/services
  - How can you get there from here?
  - How can you get results back?
- How those problems are addressed in the MonBOX Remote Monitoring Appliance

**Notes:**

- Not meant to be a sales pitch . . .
- These techniques can be implemented in many ways and in many different environments
- But I would of course enjoy receiving your feedback on this talk and on the MonBOX Remote Monitoring Appliance
  - http://monbox.com/
Real World Examples?

- Multiple locations, disjoint networks
  - Firewalls between locations
  - Links that won’t reliably pass ICMP, UDP
    * Slow, unreliable, restricted . . .

- Small, or less secure locations
  - e.g. retail chain stores

- Multiple customers
  - e.g. managing small customer office networks

Notes:

- Small customers may have unsophisticated routers/firewalls
  - And no consistent external IP address
Why Don’t You Just . . . ?

• Put a Nagios server in each location?
• Open lots of firewall ports?
• Use a global wide open VPN?
• Use mod_gearman?
Because!

- Another Nagios server requires time and money
- Security considerations may prevent VPN
- Opening firewall ports can quickly get out of hand
- You might be blocked by the OSI Network Model
  - Layers 8 (financial) and/or 9 (political)
- You might care about only a subset
  - e.g. You provide a managed device or service
- You might have no influence or power or budget

Notes:

- Small locations mean that the cost of providing and managing a separate server may not make sense
“All problems in computer science can be solved by another level of indirection”
– David Wheeler

Notes:

• I use this idea *constantly*
**Traditional: check_nrpe**

- Connect to NRPE daemon on remote host, it runs a command
  
  ```
  check_nrpe -H hosta \
  -c check_disk -w 60% -c 70% -p /
  ```

- Can allow arbitrary arguments and commands
  - Some might think that silly and less secure

- And you can hop to another host if you’re clever
  - i.e. A proxy via another layer of indirection

- Need to allow NRPE through firewall(s)

- More restricted, more obscure than SSH
Monitoring Remote Locations with Nagios

Traditional: `check_by_ssh`

- SSH to remote host, run local check, use results
  
  ```
  check_by_ssh -H hosta -- \
  \n  check_disk -w 60% -c 70% -p /
  ```

- But the check doesn't have to be a local check
  
  - The gateway host can probe a different host
  - i.e. A proxy via another layer of indirection

- And, you can get just silly:
  
  ```
  check_by_ssh -H hosta -- \
  \n  check_by_ssh -H hostb -- \
  \n  check_by_ssh -H hostc -- \
  \n  check_http -H hostd
  ```
Traditional: check_by_ssh (cont’d)

- Need to allow SSH through firewall(s)
- Want to do SNMP or other UDP checks?
  - Over a public or lossy link?
  - Use check_by_ssh to “tunnel” SNMP over unreliable links

Notes:

- There is an SNMP proxy tool out there, but SSH is easy
**Traditional: Passive Checks**

- Configure each remote machine to run checks and report back with NSCA

- Need to open a path back to Nagios
  - Same problem, different direction

- Need to install software and configure each remote device
  - Do you even have access?
  - How do you monitor switches and printers?

**Notes:**

- My preference is to avoid passive checks if I can because I think they look and act in uncommon ways
  - But I can’t always – sometimes a passive check is best
“When the going gets weird, the weird get going”
Getting Weird: SNMP Proxies

- net-snmp's `snmpd` lets you run arbitrary commands
  - Configure the exec settings
  - Commands don’t need to be local checks

- A proxy to other machines

- But . . .
  - You can’t pass arbitrary arguments
  - You need to configure the SNMP proxy machine
  - SNMP is unlikely to be the first thing allowed through a firewall
  - And it is UDP . . .

Notes:

- I never said that an SNMP proxy was one of my best ideas
Getting Weird: Web Pages

- Sometimes a remote location will have a web server
  - That you have port 80 access to
  - And some way to configure the web site

- Many web sites use PHP, CGI, or something similar
  - Why can’t you run checks through a web server?
  - PHP and CGI can run arbitrary commands

- Another proxy!

- Other protocols? SMTP?

Notes:

- There are other protocols that can be subverted
- Can you send mail?
  - Put a command payload into a message
  - Mail to an alias that pipes to a program
  - Mail the check results back to your Nagios server
- If you have access and control, you could put a daemon on some port and do anything
  - But why make a custom protocol and tool when there are existing tools that do it better?
- Use firewall port knocking to trigger something?
Getting Weirder: Phone Calls

- Got an Asterisk PBX in a remote location?
- Call it up, use DTMF to request commands
- It can call you back and report in

- OK, I never said this was a good idea

Notes:

- Trying to make the point that if you have some way to get somewhere, you can do just about anything
  - If you’re willing to be creative
Flip It Around and Passive Checks

• If you can control a machine in the remote location
  – Give it work to do
  – Cron jobs, a looping shell script, etc.

• Report back to Nagios with:
  – send_nrdp, send_nsca
  – SSH back
  – SMTP back, web calls
  – And so on . . .
**Now You’ve Got a Path, How to Use It?**

- You’ve got a mechanism to get somewhere remote
  - SSH, NRPE, etc.
- Use `mbdivert` to get Nagios to use it
- `mbdivert` diverts checks through SSH, NRPE, etc.
  - Method and destination/proxy based on hostname
  - And config file rules
- Set `$USER1$` to run `mbdivert`, or use symlinks named for plugins
  - Change what happens with barely a config change

**Notes:**

- `mbdivert` is listed in the Nagios Exchange
- or at [http://www.syonex.com/software/](http://www.syonex.com/software/)
**Reflection: Use a Third Party**

- Last year Ethan described the Nagios Reflector service
- NRDP a passive check result to a web server on the interweb
- Nagios server GETs result from the public server
- No inbound firewall rules needed
  - Outbound HTTPS is likely already allowed
- Still need to configure a remote machine somehow

**Notes:**

- `send_nrdp` to upload results
- Use `check_reflector` to retrieve the results
Reflection: Use a Complete Nagios Server

• Your reflection server could be course be a Nagios server out on the interweb

• Use an aggregation tool to give a single internal view
  – Nagios Fusion
  – Thruk and mk_livestatus

• This is not necessarily a lightweight approach
Remote Configuration

• For many of these, you still need to configure a remote machine
  – You need ongoing access to the remote device
  – Which is kind of contrary to my whole premise

• Turn reflection around
  – Send config fragments from Nagios to a third party on the interweb
  – Download work from that server into the remote location
And That’s It

- Those are my ideas on how you can monitor remote locations
- I’ll claim most of them are practical
- But you don’t need to take my word for it
  - Because I implemented many of these ideas
Quick Intro: MonBOX RMA

- MonBOX Remote Monitoring Appliance
- Built on the Raspberry Pi, runs with read only /
- Central MonBOX Management Server (MMS)
- MonBOX connects to MMS every 15 minutes
  - Gets instructions, things to monitor
  - Runs plugins from cron, or a full Nagios instance
  - Can relay results back through the MMS
  - Or directly with NRDP or NSCA (if allowed)

Notes:

- http://monbox.com/
Quick Intro: MonBOX RMA (cont’d)

- MonBOX also allows SSH, NRPE inbound
  - If the network allows access
  - Use mbdivert for ease of implementation

- Local web and console configuration

- API to interact with the MMS
  - Maintain your configs as always
  - Ship parts out to remote locations

- I would be delighted to tell you more if you’re curious

Notes:

- Please ask me, or drop me a line
- Or check out http://monbox.com/
And That’s It

• Questions?

• Need more?
  – Ask me at lunch
  – Mail me
  – Check out my sites

• Notes PDF at http://www.monbox.com/notes/

• Thank you!