Script Kiddiez Suck: V2.0

Marcus J. Ranum

<mjr@nfr.com>

Chief Technology Officer, NFR Security, Inc.

http://www.nfr.com

1

Some Background

- Black Hat briefings 2000 (Vega$) I gave a very “in your face” keynote about problems with vulnerability disclosure
  - Most of the message was lost because of its intensity and the venue
  - What did I learn?
    - It’s very hard to change people’s minds if it means reducing the amount of fun in their lives
    - Subtlety is more important than content

2

So Now What?

- I also discovered that just complaining doesn't work very well unless you can offer a solution ;)
  - `10` So I talked to friends
  - `20` I drank tequila
  - `30` I thought a lot `: GOTO 10`
  - And these are some of the results
    - With thanks to Dr. Mudge, Bruce Schneier, Vin McLellan, Lew Koch, and others...

3

Towards an Economy for Vulnerability Disclosure

Marcus J. Ranum

<mjr@nfr.com>

Chief Technology Officer, NFR Security, Inc.

http://www.nfr.com

4

Meet the Players

- The vulnerability disclosure environment has 3 players:
  - Hackers
  - Vendors
  - End Users
- Each player has their own agenda
- Each player has unique carrots and sticks that can influence them

5

Hackers

- Carrots:
  - Visibility - many disclosures are done to market hackers skills and establish a track record
- Sticks:
  - Downstream liability for actions
  - Establishing a negative reputation may hurt someday

6

Vendors

- Carrots:
  - Sales
  - Positive market image
- Sticks:
  - Prevention of sales
  - Embarrassment (note: today's disclosure economy assumes embarrassment has value but I have my doubts!)

7

Users

- Carrots:
  - Software that works better
  - Less exposure to attack by hackers
- Sticks:
  - Less reliable systems
  - Getting hacked
  - (potential) Free s/w upgrades

8

Some Challenging Thoughts

- Oral tradition in security would have it that disclosing vulnerability information is necessary in order to:
  1) Educate users
  2) Strong-arm vendors into fixing their bugs
- Nobody appears to consider the less palatable possibility:
  3) Market the person doing the disclosure

9

Some Challenging Thoughts*(cont)*

- Russ Cooper (moderator of NT Bugtraq) provides anecdotal data that about 7/10 “security alerts” are corporate or personal marketing
  – Clearly, whatever economy we derive will have to allow corporate/personal marketing for hackers, or they’ll just play by their own rules

10

Paths to Follow

- Let's look at how a vulnerability can play itself out, shall we?
  - Worst case
  - Best case
  - Ideal case

11

Evolution of A Disclosure (Worst Case)

Flaw (latent bug: not yet information)
↓
Vulnerability (now someone has discovered it)
↓
Exploit (the discoverer tests and verifies problem)
↓
Disclosed Vulnerability (the discoverer announces
↓ the problem)
Toolz (skilled hackers produce tools to exploit
↓ the problem)
Script Kiddiez (lame hackers use the toolz)

12

Evolution of A Disclosure (Best Case)

Flaw (latent bug: not yet information)

↓

Vulnerability (now someone has discovered it)

↓

Notification (the discoverer tells the vendor)

↓

Patch (the vendor announces a fix is available for the problem)

↓

Disclosed Vulnerability (the discoverer announces the problem)

↓

"Big Nothing" (what's the point in doing anything?)

13

Evolution of A Disclosure (Ideal Case)

Flaw (latent bug: not yet information)

↓

Vulnerability (now someone has discovered it)

↓

Notification (the discoverer tells the vendor)

↓

Patch (the vendor fixes the problem and keeps an eye out for signs that the flaw is discovered by someone else. If the flaw is seen in the wild the vendor pushes out the patch)

Danger: **No benefit for the hacker in this scenario!!!**

14

Paths to Follow *(redux)*

- Under the current economy the path that is best for the hacker is worst for the user and vice-versa
  - The hacker gains an inherent benefit from the shock value of the disclosure
  - This is further borne out by the fact that the worse the problem is the more newsworthy it is (benefiting the hacker)

15

Disclosing After a Patch

- Once the vendor's already released a patch, there's no benefit for the user *or* the vendor if the there is a vulnerability announcement
  - The vendor already knows
  - The user already knows
- All the hackers are doing is pounding their chests about how smart they are (lamers!)

16

Disclosing After a Patch *(cont)*

- As long as we continue to get excited about vulnerability announcements, we can continue to look forward to a flood of them as all the hackers clamor to show how smart they are
  - Even CERT is now playing by these rules
  - In an economy of *attention* the primary **product** is *noise*

17

Changing the Rules

- We have 2 choices:
  - Refine the disclosure economy so it's more predictable and less harmful to users
    - Our next topic of discussion!
  - Rewrite the rules completely
    - Fun but harder
    - Long-term more rewarding
    - Our closing topic of discussion

18

A New Economy for Disclosure

- In order to make sure that all carrots are provided and all sticks properly used, we need to add a new entity
  - A neutral third party
  - Must be vendor-neutral, externally funded, and beholden to none

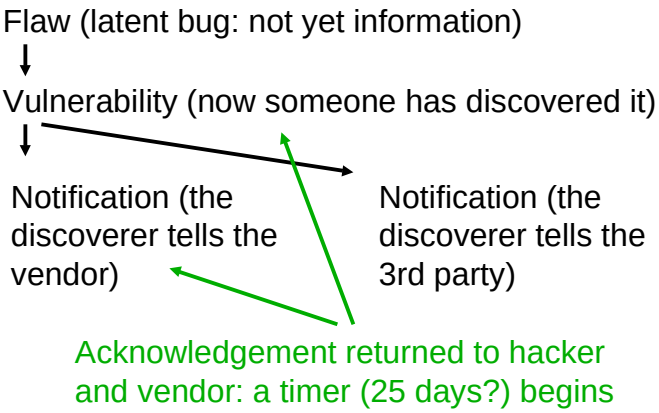
19

A New Economy for Disclosure *(cont)*

- Establish a process whereby those that properly adhere to procedures are granted recognition by the 3rd party
- Those that do not follow procedures are held in contempt; 3rd party serves as a communal memory

20

Stage 1: Identifying a Vulnerability



21

3rd Party

- The 3rd party maintains a publicly readable web site that keeps “score” including:
  - How many positive points an vendor or hacker has
  - How many negative points an vendor or hacker has
  - How many vulnerabilities in progress an vendor or hacker has

22

Stage 2: Dispatching a Vulnerability

Vendor notifies hacker and 3rd party how long they expect a fix to take or why they feel there is no vulnerability

3rd party updates website “score sheet” to indicate there is a vulnerability (nature of which is undisclosed) in product X version Y with expected fix date Z

23

Stage 3: Crediting a Vulnerability

Vendor releases fix/patch on schedule

Vendor notifies 3rd party Hacker notifies 3rd party

3rd party updates website “score sheet” to indicate that the vendor handled the problem in a positive manner, as did the hacker

24

Stage 3 Successes

- If the vendor plays by the rules they get a positive mark on their score sheet
  - This is publicly visible evidence that the vendor is responsive about security and takes it seriously
- If the hacker plays by the rules they get a positive mark on their score sheet
  - This is public evidence the hacker is smart, helpful, and plays by the rules

25

Stage 3 Failures

- If the vendor “blows off” the vulnerability the 3rd party assess them a “black mark” on their product score-sheet
- If the hacker jumps the gun and discloses the vulnerability the 3rd party assess the hacker a “black mark” on their score sheet

26

More Sticks

- Encourage audit firms (perhaps push for a FIPS?) that products which have outstanding “black marks” are not qualified for mission critical/financial/e-commerce operational deployment
- Hackers with “black marks” should be barred from employment in trusted positions (security, system admin, etc)

27

Details

- There are other details to fill in such as a review board, appeals process, etc
  - Having a neutral 3rd party does allow for many of the benefits of a disclosure environment without the more obvious disadvantages

28

Is it Going to Work?

- A more likely question is “is it going to happen?”
  - Frankly, I am unsure, because the dialog is currently being controlled by parties that benefit too much from the status quo
- Yes, it could work
  - Most users really don’t care about vulnerability details

29

Rewriting the Rules of Disclosure

- Can we?
  - Sure! Remove the key currency (marketing value of disclosure) and reduce the window of vulnerability* to near zero
  - How do we do it?
    - Patch Streaming
    - Other ways?

* See Schneier’s October issue of Crypto-Gram

30

Patch Streaming

- Prediction: this will be all over in 5 years
  - Software can self-update in the event of a security flaw
  - Cooperative control (settable by user) with remote download (controllable by vendor)
  - Provides the vendor huge marketing benefits (positive customer touch, s/w maintenance revenue) and the customer easier/faster upgrades and security

31

Patch Streaming *(cont)*

If there is a critical bug or security flaw in this software, I should:

a) Ignore it and notify an administrator

b) Cease operation immediately and notify the administrator

c) Keep operating in reduced capacity

d) Attempt to automatically upgrade myself to a new release

32

Patch Streaming *(cont)*

- To implement patch streaming we need existing tools:
  - Secure web servers
  - Signature of code
  - PKI/certificates
- None of this is advanced rocket science: antiviral programs and browsers do it already

33

Patch Streaming *(cont)*

- In a patch streaming environment:
  - There is little benefit to the hacker to disclose anything
    - On the contrary: hackers will hoard their techniques because as soon a technique is known it becomes ineffective
  - There's no point in making announcements
  - The vulnerability window closes very quickly

34

Summary

- I hope this talk has been a bit more positive and productive than my last one
  - Certainly it sounds more friendly ;)
  - Read between the lines and you'll see I'm showing the community how to pull the teeth from all the grey hat hackers
- Thanks for your time and attention!

35

References

- Bruce Schneier's Crypto-Gram article on controlling vulnerability exposure by time (recommended reading)
  - http://www.counterpane.com/crypto-gram-0009.html
- Various mjr-oid rants on the topic:
  - http://www.ranum.com/pubs

36