Dealing with Adversarial Relationships in Information Security

Daniel Crowley, Head of Research

X-Force Red

IBM

Common Misconceptions

#1: We Create Problems

We do not create vulnerabilities, only find them

- QA doesn't create bugs, do they?
- Vulns come from development, design, and inheritance

We are not making more work

- We identifying where work already needs doing

#2: We Overstate Severity

If we're not careful, this may be true

- Vuln scanners and pen testers are incentivized to inflate severity
- We may feel personal attachment to our findings

If everything is Critical, nothing is

- Severity exists to allow prioritization of limited resources
- Impact must be tempered by pre-requisites (e.g. BEAST)

Have a real, thought-out framework for severity

#3: We Are Incompetent

"If you're a real hacker why do you need"

- Internal network access
- Credentials
- Documentation

A bug you don't really understand may look like bullshit

- SSRF

Be ready to explain your job / findings

#4: We Want to Get People Fired

Most of the time, this isn't true

Thankfully, we aren't incentivized to get people fired

#5: We Are Trying to Extort People

Most of us aren't

Mostly a problem in vulnerability disclosure

If you don't understand someone's motivation, you may assign them one

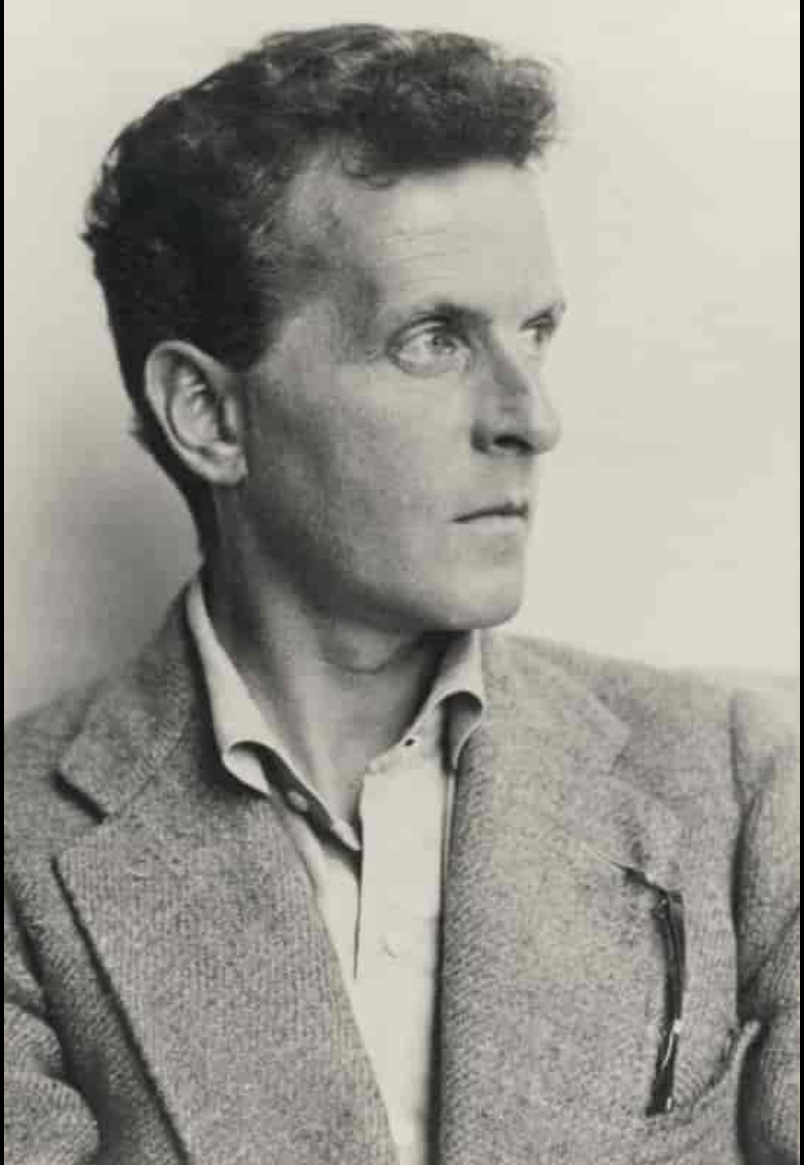
- Alluding to your motivation helps

Language-Games

Ludwig Wittgenstein

German philosopher

Focused on communication and miscommunication



What is a Language-Game?

Language applied to some mode of life

- A verbal protocol with a goal

Goal can be many things including:

- Education
- Conveyance of facts and figures
- Instruction
- Comfort
- Recreation / Entertainment

A simple, strict language-game

A surgeon has a series of command words they can speak

Every command word is first repeated by the assistant as acknowledgement

Some of these are tool names (scalpel, forceps)

The assistant provides the tool

Some of these are processes (wipe, pressure)

The assistant performs the associated action

A deceptive language-game

When someone "wants to debate" you on the Internet

And they're "just asking questions"

And they want to "consider all ideas equally"

It may be that they are not playing the Collaborative Mutual Education game

They may be playing the Promote Bad Ideas game

When language-games clash

You may share a fictional story, playing the Storytelling game

Your story may mistakenly be received as History Lessons

You may be playing the Relax With A Friend game

But if you ask someone if they want to watch Netflix and chill

They might think you're playing a very different game

Infosec language games

Convey Risk

Communicate Bug Details

Security Education

Share Threat Information

Get Management Support

Prove I'm a Good Hacker

Some adversarial games

Prove that I'm Not Stupid

Keep My Job

Reduce My Workload

Meet the Deadline

Keep Customers / Contract

Pass the Audit

Identifying and Resolving Adversarial Games

Prove that I'm Not Stupid

Ego may be hurt when mistakes are pointed out

- Nobody likes having their baby called ugly
- There are lots of ugly babies nonetheless

Game player will try to prove they're smart or you're stupid

When:

- Disclosing issues to the person responsible for them

Possible tells:

- Downplay / denial of bugs they are responsible for
- Condescension
- Unnecessary displays of knowledge
- Pedantic nitpicking

Prove that I'm Not Stupid - Fix

Provide an example of good security along with bad

- Not your job, but easier than fighting ego

Give an out like "nobody gets this right first try"

Don't do things like putting people's passwords in reports

- Don't make people feel stupid and you won't have to deal with this

Find something to agree with them on

Keep My Job

Some people may fear for their job

- At times rightfully so

In most cases, vulns aren't a job-ender

Game player will try to maximize their perceived worth

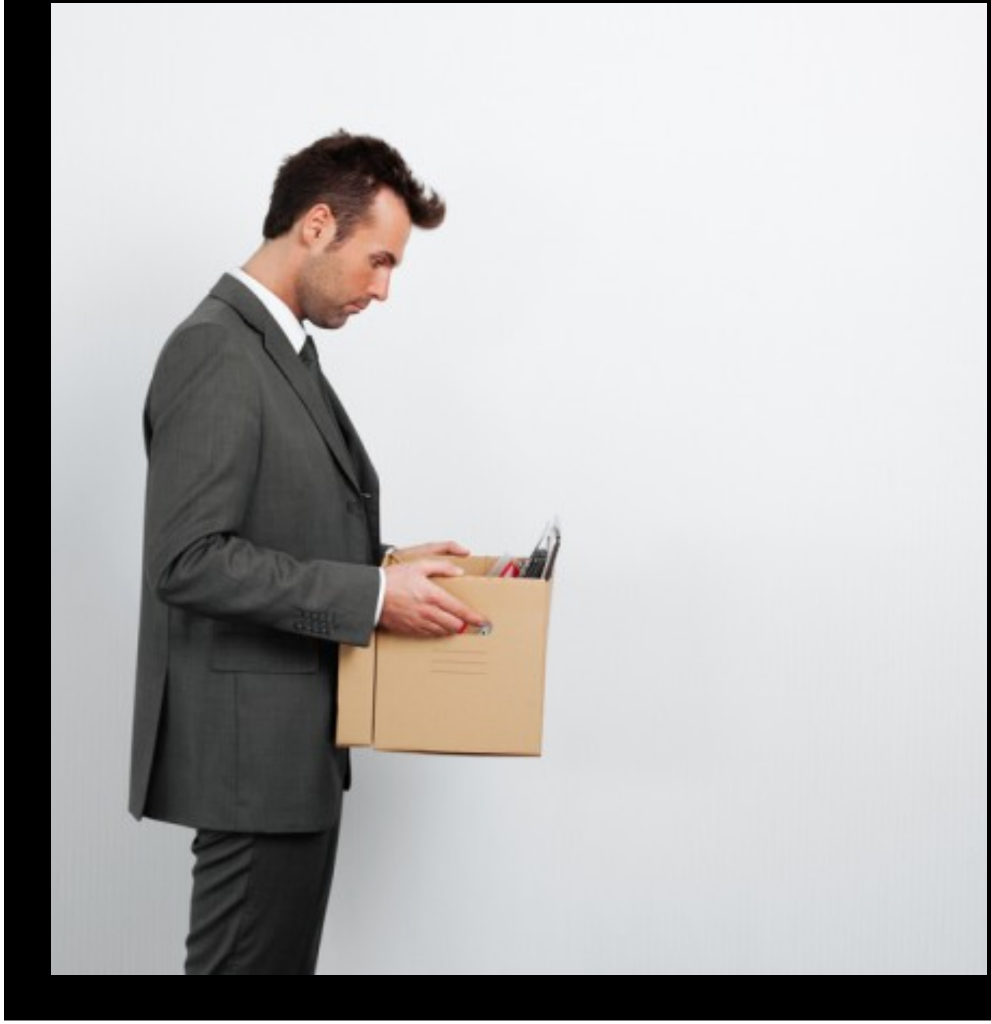
When:

- Disclosing issues to the person responsible for them with management present

Possible tells include:

- Blame shifting
- Denial / downplay of all bugs they may have created
- Attacks on your competence / character
- Reminders of their worth / effort

Keep My Job - Fix



If it's not malicious or criminally negligent

- "It's an easy mistake to make"

Remind that even the best programmers make bugs

- Security bugs are a subclass of bugs

"Security is a process"

If possible, do not include managers on the call

Reduce My Workload

Some developers are overworked

- Others are not overworked but want to do less work anyway
- Still others are required to fix security bugs free by contract

Won't be this if game player isn't responsible for addressing issue!

Game player will try to minimize response effort

When:

- Disclosing to someone responsible for addressing the issue

Possible tells include:

- Asking if existing controls are enough
- Exaggerating cost / effort of fix
- Shifting responsibility
- Denial / downplay of bugs they are responsible for
- Throwing red tape

Reduce My Workload - Fix

Discuss solutions, especially low-effort but effective ones

- "There are plug-in frameworks built to mitigate CSRF"
- "You can use NaCl and not have to create your own cryptosystem"
- "Your web framework has a drop-in authc/authz system you can use"

Refute bug denials

If needed, include their manager on the call

Meet the Deadline

Fixing bugs takes time

Some devs get too little time to work

- Speed to market means more profit

Some security tests get thrown in right before release allowing no time for security fixes

More likely to be played by management

Can't be this with no deadline

Game player will try to reduce fix time

When:

- Disclosing issues

Possible tells include:

- Denial of only deadline blocking bugs
- Asking questions about how long to fix
- Can it be fixed after go-live
- Throwing red tape

Meet the Deadline - Fix

- Suggest quickly implemented solutions for hard to fix issues
- Remind them that severity rankings exist to aid prioritization
- Throw the person who decided to schedule security testing a week before go-live under the bus

Keep Customers / Contract

Some people see security and usability as a dichotomy

– It isn't, fixing a buffer overflow doesn't change UX

Some people want to deny bug existence to avoid damaging company image

– These people are shitty

Game player will minimize bugs and maximize their own value

When:

– Disclosing issues, usually to mgmt / sales-involved people

– Bug readout call with contract buyer and seller

Possible tells include:

– Denial / downplay of all bugs

– Mentioning customer impact of fixing bugs

– "Feature not bug"

Keep Customers / Contract - Fix

Point out false dichotomy between usability / security

If doing coordinated vulnerability disclosure:

- Remind them bugs are going public either way
- They choose how to respond to early warning about the bugs
- They can be the hero or the villain (don't actually say this)

Point out that customers hate having their identity stolen more than UX changes

Pass the Audit

Some security testing is done for compliance

- PCI
- HIPAA
- Govt-mandated

Audits are not fun

People usually want to get back to business as usual

Game player will focus on passing the audit as cleanly as possible

When:

- Test readout as part of audit

Potential tells include:

- Denial of only audit blocking bugs
- Bringing up audit language / compliance terms

Pass the Audit - Fix

Remind them you don't decide pass/fail

- Unless you do

Punish For the Outage

Outages cost money

They're also a risk associated with testing in prod

If you cause one, you will have people angry at you

When:

- You took something down by mistake

Tells:

- Pretty obvious

Punish For the Outage - Fix

Reframe the situation

- "I'm so glad we found this problem; all it took was one laptop and a mistake"
- "Can you imagine if someone was TRYING to do this?"
- People have thanked me at the end of a call where they had planned to burn me at the stake

Make sure you mention you're aware of and regret the outage

Ask if testing can be switched to a non-prod environment (if applicable)

Remember:

- It isn't really your fault, you weren't trying to knock it over
- Every seasoned security tester has at least one of these stories

General Advice

General tips – Identifying games

Asking for confirmation can help identification

- Works well for Pass the Audit
- Doesn't work for Prove I'm not Stupid

Learn about the dynamics of the situation

- Who are the players?
- What do they stand to gain / lose?

Use process of elimination to narrow possibilities

- Can't be Meet the Deadline with no deadline

General tips – Defending bugs

Be honest with severity rankings

–If every bug is High / Crit, none of them are

–Verbose error messages aren't as bad as SQLi

Create air-tight bug reports

–Give evidence of the bug's existence and impact

–Severity should consider impact AND pre-requisites

General tips – Defending bugs

Handling arguments over low-severity bugs

- Remind that Info finding aren't bugs, just information!
- "I agree, and this was considered in the report"
- Hand-in-toaster rule
- "This was included for completeness, we do not only report the worst bugs"

General tips – Defending bugs

Remind them:

- You are paid to report based on your expertise
- You do not make final business decisions on what, when or how to fix

If doing vendor disclosure:

- Remind them that the bugs are going public even if unfixed
- The disclosure is being delayed out of courtesy
- They are free NOT to fix the bugs if they choose

General tips – Handling character / competence attacks

The technical validity and severity of information is not changed by:

- Your intelligence
- Your experience
- Your character

Deflect any personal attacks by shifting the conversation back to the bugs

- This exposes the attacks, and thus the attackers, as unprofessional
- This also
  - Highlights your own professionalism
  - Quickly puts down this tactic